

1.1.1 CURRICULUM PLANNING AND IMPLEMENTATION

S.NO	CONTENTS	PAGE NO.
INSTITUTE LEVEL		
1.	Academic Calendar	3
2.	Overall Time-Table preparation – Classwise hour allocation plan including Value Addition Initiatives & Summary	8
DEPARTMENT LEVEL		
3.	Subject Allocation Circular, Staff willingness, Competency Matrix	10
4.	Subject allocation classwise	16
5.	Staff workload	18
6.	Syllabus Completion Status verification	20
7.	Internal Assessment Result review(consolidated statement)	23
8.	Revision /Coaching Classes plan	30
9.	Test report (covering Corrective, Preventive action)	38
COURSE IN-CHARGE LEVEL		
10.	Individual time table	39
11.	Student Name list	40
12.	Log Book	41
13.	Syllabus	53
14.	Course plan	55
15.	Question Bank (Sample pages)	61
16.	Unitwise notes (Sample pages –Handwritten, e-Material)	82
17.	Internal assessment question paper, answer key (Assessment test-I, II, Model examination)	108
18.	Internal assessment mark statement (Coursewise statement & Consolidated statement)	122
19.	Format A (for assignments) & Sample	144

S.NO	CONTENTS	PAGE NO.
20.	Proofs for teaching Aids / methodology ➤ CD – PPT, Videos, CBTs, NPTEL ➤ Tutorial Sheets ➤ Report for Industrial Visits / Demo sessions / Guest Lectures ➤ Identified PCE skills and execution proofs	157
21.	Format B (for content beyond syllabus)	169
22.	Support to slow learner ➤ Identified slow learner list ➤ Remedial class plan & attendance ➤ Supporting documents	172
23.	Support to advanced learner ➤ Identified advanced learner list ➤ Promotional activity details ➤ Supporting documents	
24.	Sample assessment papers (sample) for all assessments.	218
25.	University question paper	239
26.	Review Sheet – Experience summary by Course-incharge	241

KINGS COLLEGE OF ENGINEERING

Academic Calendar Academic Year 2019-2020 (Odd Semester)

JUNE & JULY 2019

DATE	DAY	EVENTS
20.06.19	Thursday	College Reopens for II,III, IV Year UG / II Year PG
21.06.19	Friday	
22.06.19	Saturday	Holiday
24.06.19	Monday	
25.06.19	Tuesday	
26.06.19	Wednesday	
27.06.19	Thursday	Class Committee Meeting I for II, III & IV Year / II Year PG
28.06.19	Friday	Class Committee Meeting I for II, III & IV Year
29.06.19	Saturday	Holiday
01.07.19	Monday	Submission of Status of Study Material to Principal by HODs Submission of IHT Report to Principal by HODs
02.07.19	Tuesday	Submission of Class Committee Meeting I Minutes to Principal by HODs
03.07.19	Wednesday	Staff Council Meeting - I
04.07.19	Thursday	
05.07.19	Friday	IQAC Meeting –I Professional Society Activities
06.07.19	Saturday	Holiday
08.07.19	Monday	
09.07.19	Tuesday	Submission of IQAC Meeting I Minutes by IQAC Coordinator
10.07.19	Wednesday	Submission of DRM I Minutes by HODs to IQAC Coordinator
11.07.19	Thursday	
12.07.19	Friday	
13.07.19	Saturday	Working Day
15.07.19	Monday	Submission of DRC Meeting I Minutes to Principal Submission of Class Test I Marks Statement for II, III & IV Year UG
16.07.19	Tuesday	
17.07.19	Wednesday	Submission of Assessment Test I Question Papers to CCE office
18.07.19	Thursday	Staff Appraisal
19.07.19	Friday	Submission of Syllabus Completion Report by HODs Submission of Assignment I Status to Principal by HODs
20.07.19	Saturday	Holiday
22.07.19	Monday	Assessment Test I Commences for UG II, III, IV Year & PG II Year
23.07.19	Tuesday	
24.07.19	Wednesday	Assessment Test I Ends for PG II Year
25.07.19	Thursday	
26.07.19	Friday	II PG Project Review I
27.07.19	Saturday	Assessment Test I Ends for UG II, III, IV Year
29.07.19	Monday	
30.07.19	Tuesday	
31.07.19	Wednesday	

NO. OF WORKING DAYS: 32

KINGS COLLEGE OF ENGINEERING

Academic Calendar Academic Year 2019-2020 (Odd Semester)

AUGUST 2019

DATE	DAY	Events
01.08.19	Thursday	Submission of Assessment Test I Result Analysis by HODs & Review Meeting With Principal Submission of Assessment Test I Answer Scripts to CCE office (MECH, EEE)
02.08.19	Friday	Submission of Assessment Test I Answer Scripts to CCE office (CSE, CIVIL, ECE) Professional Society Activities
03.08.19	Saturday	Working day - Monday Order Counseling I for II, III & IV Year
05.08.19	Monday	Class Committee Meeting II for II, III & IV Year/ II Year PG
06.08.19	Tuesday	Class Committee Meeting II for II, III & IV Year
07.08.19	Wednesday	Staff Council Meeting II Submission of Counseling I Report
08.08.19	Thursday	Submission of Class Committee Meeting II Minutes to Principal by HODs
09.08.19	Friday	IQAC Meeting -II
10.08.19	Saturday	Holiday
12.08.19	Monday	Bakrid - Holiday
13.08.19	Tuesday	Submission of Class Test II Marks statement for II, III & IV Year UG Submission of DRM II Minutes by HODs to IQAC Coordinator Submission of IQAC Meeting II Minutes by IQAC Coordinator
14.08.19	Wednesday	Submission of Assessment Test II Question Papers to CCE office
15.08.19	Thursday	Independence Day - Flag Hoisting Ceremony
16.08.19	Friday	Submission of DRC Meeting II Minutes to Principal Submission of Syllabus Completion Report by HODs Submission of Assignment II Status to Principal by HODs
17.08.19	Saturday	Working Day
19.08.19	Monday	Assessment Test II Commences for UG II, III, IV Year & PG II Year
20.08.19	Tuesday	
21.08.19	Wednesday	Assessment Test II Ends for PG II Year
22.08.19	Thursday	
23.08.19	Friday	II PG Project Review II
24.08.19	Saturday	Assessment Test II Ends for UG II, III, IV Year
26.08.19	Monday	
27.08.19	Tuesday	
28.08.19	Wednesday	
29.08.19	Thursday	Symposium - CSE, ECE, EEE Submission of Assessment Test II Result Analysis by HODs & Review Meeting With Principal Submission of Assessment Test II Answer Scripts to CCE office (CIVIL, MECH)
30.08.19	Friday	Symposium - CIVIL, MECH Submission of Assessment Test II Answer Scripts to CCE office (CSE, ECE, EEE)
31.08.19	Saturday	Holiday

NO. OF WORKING DAYS: 23

KINGS COLLEGE OF ENGINEERING

Academic Calendar Academic Year 2019-2020 (Odd Semester)

SEPTEMBER 2019

DATE	DAY	Events
02.09.19	Monday	Vinayagar Chaturthi - Holiday
03.09.19	Tuesday	
04.09.19	Wednesday	Staff Council Meeting III
05.09.19	Thursday	Teachers day
06.09.19	Friday	IQAC Meeting -III Professional Society Activities
07.09.19	Saturday	Working day - Tuesday Order Counseling II for II, III & IV Year
09.09.19	Monday	Submission of IQAC Meeting III Minutes by IQAC Coordinator
10.09.19	Tuesday	Muharram - Holiday
11.09.19	Wednesday	Submission of DRM III Minutes by HODs to IQAC Coordinator Submission of Counseling II Report Class Committee Meeting III for II, III & IV Year/ II Year PG
12.09.19	Thursday	Class Committee Meeting III for II, III & IV Year
13.09.19	Friday	Zeroth Project review for Final year UG Submission of Syllabus Completion Report by HODs
14.09.19	Saturday	Working Day
16.09.19	Monday	Submission of Class Committee Meeting III Minutes to Principal by HODs Submission of DRC Meeting III Minutes to Principal Revision classes Commences for UG II, III, IV Year - Phase I
17.09.19	Tuesday	
18.09.19	Wednesday	Submission of Model Exam Question Papers to CCE Office
19.09.19	Thursday	
20.09.19	Friday	Revision classes Ends for UG II, III, IV Year - Phase I
21.09.19	Saturday	Holiday
23.09.19	Monday	Model Exam: Theory 1 for UG II, III, IV Year & PG II Year
24.09.19	Tuesday	Model Exam: Theory 2 for UG II, III, IV Year & PG II Year
25.09.19	Wednesday	Model Exam: Theory 3 for UG II, III, IV Year & PG II Year
26.09.19	Thursday	Model Exam: Theory 4 for UG II, III, IV Year
27.09.19	Friday	Model Exam: Theory 5 for UG II, III, IV Year II PG Project Review III
28.09.19	Saturday	Model Exam: Theory 6 for UG II, III, IV Year
30.09.19	Monday	Revision classes Commences for UG II, III, IV Year - Phase II

NO. OF WORKING DAYS: 22

KINGS COLLEGE OF ENGINEERING

Academic Calendar Academic Year 2019-2020 (Odd Semester)

OCTOBER 2019

DATE	DAY	Events
01.10.19	Tuesday	
02.10.19	Wednesday	Gandhi Jayanthi - Holiday
03.10.19	Thursday	Staff Council Meeting IV Submission of Model Exam Result Analysis by HODs & Review Meeting With Principal Submission of Model Exam Answer Scripts to CCE office (MECH, EEE)
04.10.19	Friday	IQAC Meeting -IV Submission of Model Exam Answer Scripts to CCE office (CSE, CIVIL, ECE)
05.10.19	Saturday	Holiday
07.10.19	Monday	Ayutha Pooja - Holiday
08.10.19	Tuesday	Vijaya Dasami - Holiday
09.10.19	Wednesday	Submission of IQAC Meeting IV Minutes by IQAC Coordinator
10.10.19	Thursday	Submission of DRM IV Minutes by HODs to IQAC Coordinator
11.10.19	Friday	Revision classes Ends for UG II, III, IV Year – Phase II
12.10.19	Saturday	Holiday
14.10.19	Monday	Revision classes Commences for UG II, III, IV Year – Phase III Submission of Subject Allocation Report for next semester
15.10.19	Tuesday	Submission of DRC Meeting IV Minutes to Principal
16.10.19	Wednesday	Stock Verification & ISO Internal Audit Commences
17.10.19	Thursday	
18.10.19	Friday	
19.10.19	Saturday	Revision classes Ends for UG II, III, IV Year – Phase III Last Working day – Wednesday Order
21.10.19	Monday	Commencement of Practical Examinations
22.10.19	Tuesday	
23.10.19	Wednesday	
24.10.19	Thursday	
25.10.19	Friday	Stock Verification & ISO Internal Audit Ends
26.10.19	Saturday	
28.10.19	Monday	
29.10.19	Tuesday	
30.10.19	Wednesday	Submission of Report on Stock Verification, ISO Internal Audit by coordinators
31.10.19	Thursday	

NO. OF WORKING DAYS: 22

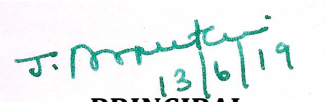
KINGS COLLEGE OF ENGINEERING

Academic Calendar Academic Year 2019-2020 (Odd Semester)

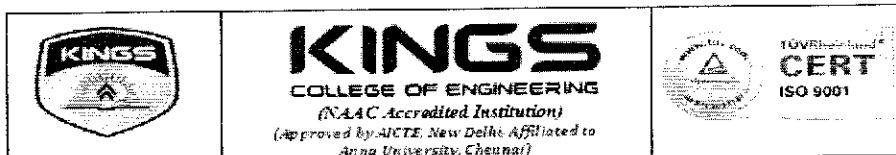
NOVEMBER 2019

DATE	DAY	Events
01.11.19	Friday	Last Date for submission of LM, QB for next semester
02.11.19	Saturday	Holiday
04.11.19	Monday	
05.11.19	Tuesday	
06.11.19	Wednesday	Commencement of End Semester Examinations Staff Council Meeting V
07.11.19	Thursday	
08.11.19	Friday	IQAC Meeting – V
09.11.19	Saturday	
11.11.19	Monday	Submission of DRM V Minutes by HODs to IQAC Coordinator
12.11.19	Tuesday	Submission of IQAC Meeting V Minutes by IQAC Coordinator
13.11.19	Wednesday	
14.11.19	Thursday	
15.11.19	Friday	Submission of DRC Meeting V Minutes to Principal
16.11.19	Saturday	Holiday
18.11.19	Monday	
19.11.19	Tuesday	
20.11.19	Wednesday	
21.11.19	Thursday	
22.11.19	Friday	
23.11.19	Saturday	
25.11.19	Monday	
26.11.19	Tuesday	
27.11.19	Wednesday	
28.11.19	Thursday	
29.11.19	Friday	
30.11.19	Saturday	

NO. OF WORKING DAYS: 24


PRINCIPAL

CC: Secretary/ CEO
VP/HODs/ AO
DW-Hostels/Transport/Canteen/HS-GH

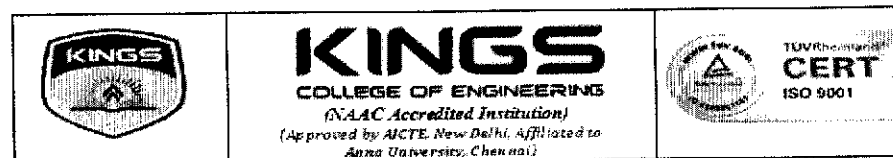


ACADEMIC YEAR 2019– 20 ODD SEMESTER GUIDELINES FOR TIMETABLE PREPARATION

- College Timing is continued as 9.15 AM to 4.30 PM (8 Hrs / Day) (45 min / Hr).
- Lecture Hours
 - Maximum 5 to 6 Hrs allocated for tough Subjects (Credit 4 or Tutorial or credit 3 (based on its difficulty)) and 3 to 4 Hrs allocated for remaining subjects (Credit 3).
 - Toughest subject is selected by concern HOD based on the results obtained in the previous year.
 - 5th hour will be utilized based on HOD recommendation
 - Tutorial Subjects / Elective Subjects must be mentioned in timetable itself
- Lab Hours
 - Allocate 4 hrs for II & III Years (**Regulation 2017**)
 - Allocate 3 hrs for IV Year (**Regulation 2013**)
 - All the lab sessions should be handled at afternoon session only.
- Excess Hours
 - Excess Hours will be implemented in Saturdays
 - IV Year – GATE – 1 Hr / Week
 - IV Year – Project Phase – I – 1 Hr/Week
- Training & Placement Hour
 - Allocate 2 Hrs / week to all department students.
 - II year – Soft Skill – 2 Hrs / Week
 - III year – Aptitude – 2 Hrs / Week
 - IV Year
 - Soft skill – 2 Hrs / Week
 - Aptitude – 1 Hr / Week
- Weekly one hour is allocated for Library & Internet
- Same Timings will be followed for I Year.
- Guidelines are also same to PG

S. P. R. 10/06/2019
OVERALL TIMETABLE COORDINATOR

J. R. R. 10/6/2019
PRINCIPAL



ACADEMIC YEAR 2019 – 20 ODD SEMESTER

TIME TABLE PREPARATION – CREDIT BASED ANALYSIS

DEPT	II YEAR				III YEAR				IV YEAR			
	C-4	C-3	C-2	total	C-4	C-3	C-2	Total	C-4	C-3	C-2	Total
CIVIL	1 1*5=5	5 5*4=20	3 C-1→2 2*4=8	35	1 1*5=5	5 5*4=20	3 3*4=12	37	-	6 6*4=24	2 2*3=6	30
CSE	2 2*5=10	3 3*4=12	4 C-1→2 3*4=12	36	1 1*5=5	5 5*4=20	3 3*4=12	37	-	6 6*4=24	2 2*3=6	30
ECE	2 2*5=10	4 4*4=16	3 C-1→2 2*4=8	36	1 1*5=5	5 5*4=20	3 3*4=12	37	-	6 6*4=24	2 2*3=6	30
EEE	1 1*5=5	5 5*4=20	2 2*4=8	33	-	6 6*4=24	3 C-1→2 2*4=8	34	-	6 6*4=24	2 C-1→2 1*3=3	29
MECH	3 3*5=15	2 2*4=8	3 C-1→2 3*4=12	37	1 1*5=5	4 4*4=16	3 3*4=12	33	-	6 6*4=24	3 C-1→1 2*3=6	31

Total Hours (Except IV Year) : 5 * 8 = 40 Hrs

C-4 – 5 Hrs

C-3 – 4 Hrs

C-2 – 2 Hrs

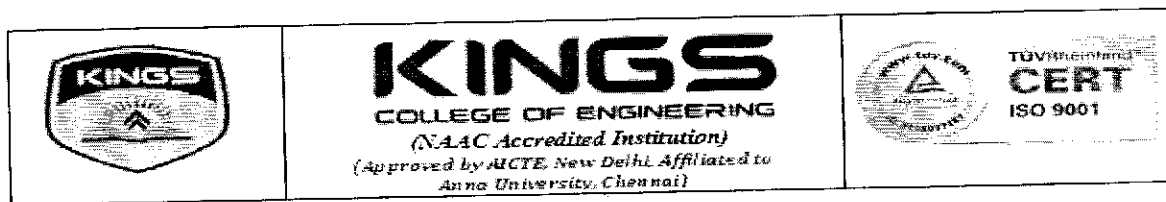
- Lib / Net : 1 Hr / Week
- Training & Placement : 2 Hrs / Week

S. P. R.
3/6/19

OVERALL TT

J. M. M. M.
03/6/2019

PRINCIPAL



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ACADEMIC YEAR 2019 - 2020 (ODD SEM)

CIRCULAR

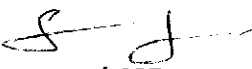
Date: 01.04.19

Staff members are requested to mention their willingness to opt Theory & Laboratory papers for the forthcoming academic year **2019-2020 ODD Semester**.

- Senior & experienced faculties shall prefer to opt Tough / Problematic paper thereby helping in securing good results.
- Kindly opt UG Papers only in willingness chart
- We will convene department meeting to finalize papers on **03.04.19** at CSE LAB II by **3.00 P.M.**

Encl:

1. Competency Matrix
2. Willingness Chart


HOD / CSE



KINGS
COLLEGE OF ENGINEERING
(NAAC Accredited Institution)
(Approved by AICTE, New Delhi, Affiliated to
Anna University, Chennai)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ACADEMIC YEAR 2019 – 2020 (ODD SEMESTER)

WILLINGNESS CHART FOR SUBJECT ALLOCATION

SNO.	STAFF NAME	SUB 1	SUB 2	SUB 3	LAB 1	LAB 2	CDC
1.	Dr.S.M.UMA HOD/CSE	OOAD	MPmc	Computer Network	—	—	—
2.	Ms.K.ABHIRAMI (HEAD IQAC)	Data Structures (CSE)	Open Elective	CS6007 IRT			
3.	Ms.S.PUVANESWARI	Data Structures (CSE)	Theory of Computation	Computer Networks	Data Structures Lab	PSPP Lab	—
4.	Ms.B.SANGEETHA	OOAD	OOPS	GUI			
5.	Mr.K.RAJESH	CS6007 IRT	CS8591 CN	Open Elective			

SNO.	STAFF NAME	SUB 1	SUB 2	SUB 3	LAB 1	LAB 2	CDC
6.	Mr.R.RANITHA	CS8592 OoAD	CS6703 GCC	CS6004 Cyber Forensics	CS8582 OoAP lab	CS8581 N/W lab	
7.	Mr.S.RAJARAJAN	CS6701 CNS	CS6004 CF	CS8391 DS	CS6711 SL	CS8381 DSL	
8.	Mr.D.SIVAKUMAR	CS8392 OOPS	CS8391 DS	CS6701 CNS	CS8383 OOPS LAB	CS8381 DS LAB.	
9.	Ms.P.NALAYINI	CS8591 CN	CS8592 OoAD	CS6007 IR	CS8581 N/W lab	88582 OoAD lab	
10.	Ms.R.SUGANTHA LAKSHMI	CS8591 CN	open elective	CS6007 IRT	CS8581 N/W Lab	CS8381 DS Lab.	
11.	Mr.R.SRIRAMKUMAR	CS6703 GCC	CS6004 CF	CS8391 DS	CS8381 DSL lab	CS8582 OoAP lab	
12.	Mr.M.ARUN	EC8393 DSVC II-BE	open elective	GE8151 PSPP	CS8383 OOPS lab	EC8381 DSL lab II-BE	
13.	Ms.G.CHANDRA PRABA	CS8391-DS (CSE)	CS8591- CN	CS6007 - IRT	CS8381 DS Lab	CS8581 - CN Lab	

A. Pur

f



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ACADEMIC YEAR 2019 – 2020 (ODD SEMESTER)

COMPETENCY MATRIX FOR SUBJECT ALLOCATION

SNO	STAFF NAME		S.M.UMA	K.ABHIRAMI	S.PUVANESWARI	B.SANGEETHA	K.RAJESH	R.RANITHA	S.RAJARAJAN	D.SIVAKUMAR	P.NALAYINI	R.SUGANTHALAKSHMI	R.SRIRAMKUMAR	M.ARUN	G.CHANDRA PRABA
I YEAR															
1.	GE8151	Problem Solving & Python Programming		***	**					*			x x	**	
2.	GE8161	Problem Solving & Python Programming Lab											x		
II YEAR															
3.	CS8391	Data Structures		***	**				*	***			**		*
4.	CS8392	Object Oriented Programming				**			*	***					**
5.	CS8381	Data Structures Lab		***	**					***			**		*
6.	CS8383	OOPs Lab								***				**	*

SNO	STAFF NAME		S.M.UMA	K.ABHIRAMI	S.PUVANESWARI	B.SANGEETHA	K.RAJESH	R.RANITHA	S.RAJARAJAN	D.SIVAKUMAR	P.NALAYINI	R.SUGANTHALAKSH	R.SRIRAMKUMAR	M.ARUN	G.CHANDRA PRABA
	SUBJECT														
III YEAR															
7.	CS8591	Computer Networks		*	*		**			**	*	*			*
8.	EC8691	Microprocessors & Microcontrollers	**		*					**	*	*			*
9.	CS8501	Theory of Computation			**								**		
10.	CS8592	Object Oriented Analysis & Design	**		*	*		*		*	*				*
11.		Open Elective			*	*		*		*	*				*
12.	EC8681	Microprocessors & Microcontrollers Laboratory										*		*	*
13.	CS8582	Object Oriented Analysis & Design Laboratory						*		*			**		
14.	CS8581	Networks Laboratory						*		*		*			
IV YEAR															
15.	CS6701	Cryptography & Network Security		*	*	*	*		*	*					
16.	CS6703	Grid and Cloud Computing			*	*		*		*	*				
17.	CS6004	Cyber Forensics (Elective)	*	*	*	*		*	*		*	*	*		*

SNO	STAFF NAME SUBJECT		S.M.UMA	K.ABHIRAMI	S.PUVANESWARI	B.SANGEETHA	K.RAJESH	R.RANITHA	S.RAJARAJAN	D.SIVAKUMAR	P.NALAYINI	R.SUGANTHALAK	R.SRIRAMKUMAR	M.ARUN	G.CHANDRA PRABA
18.	CS6007	Information Retrieval Techniques (Elective)		*			**					*			*
19.	CS6711	Security Laboratory							***						
20.	CS6712	Grid and Cloud Computing Laboratory													
Other Dept Paper															
21.	EC8393	Data Structures using C (II ECE)			**									**	
22.	EC8381	Data Structures Lab (II ECE)			**									**	
23.	CS8392	Object Oriented Programming			*										
24.	CS8383	OOPs Lab													
25.															

* Willing to handle

** Capable of Handling

*** Expertise

Note: Mention any one of CDC


PREPARED BY

(Mrs.S.Puvaneswari AP/CSE)


APPROVED BY

(Dr.S.M.Uma HOD/ CSE)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ACADEMIC YEAR 2019 – 2020 (ODD SEMESTER)

CLASSWISE SUBJECT ALLOCATION RECORD

II YEAR

SUB CODE	SUBJECT NAME	STAFF NAME
MA8351	Discrete Mathematics	Ms.S.Geetha
CS8351	Digital Principles and System Design	Mr.S.Rajarajan
CS8391	Data Structures	Ms.K.Abhirami
CS8392	Object Oriented Programming	Dr.D.Sivakumar
EC8395	Communication Engineering	Mr.T.Jeyaseelan
CS8381	Data Structures Laboratory	Mr.S.Rajarajan & Ms.K.Abhirami
CS8382	Digital Systems Laboratory	Mr.R.Sathyaraj / Mr. W. Newton David Raj
CS8383	Object Oriented Programming Laboratory	Dr.D.Sivakumar & B.Sangeetha
HS8381	Interpersonal Skills/ Listening & Speaking	Mr.P.Rajeshwaran

III YEAR

SUB CODE	SUBJECT NAME	STAFF NAME
MA8551	Algebra and Number Theory	Mr.G.Jeyakrishnan
CS8591	Computer Networks	Mr.K.Rajesh
EC8691	Microprocessor & Microcontroller	Dr.S.M.Uma
CS8501	Theory of Computation	Ms.S.Puvaneswari
CS8592	Object Oriented Analysis & Design	Ms.R.Ranitha
OMF551	Product Design and Development	Ms.R.Suganthalakshmi
EC8681	Microprocessor & Microcontroller Lab	Ms.S.Puvaneswari & Dr.S.M.Uma
CS8582	Object Oriented Analysis & Design Lab	Ms.R.Ranitha & Mr.K.Rajesh
CS8581	Networks Lab	Ms.R.Suganthalakshmi & Ms.S.Puvaneswari

IV YEAR

SUB CODE	SUBJECT NAME	STAFF NAME
CS6701	Cryptography and Network Security	Dr.D.Sivakumar
CS6702	Graph Theory and Applications	Dr.R.Suresh
CS6703	Grid and Cloud Computing	Ms.B.Sangeetha
CS6704	Resource Management Techniques	Dr.G.Shankara Kalidoss
CS6004	Cyber Forensics	Mr.R.Sriramkumar
CS6007	Information Retrieval	Ms.P.Nalayini
CS6711	Security Laboratory	Ms.P.Nalayini & Dr.D.Sivakumar
CS6712	Grid and Cloud Computing Lab	Mr.R.Sriramkumar & Ms.B.Sangeetha

PG - I YEAR

SUB CODE	SUBJECT NAME	STAFF NAME
MA5160	Applied Probability and Statistics	Dr.G.RamyaArokiyaMary
CP5151	Advanced Data Structures and Algorithms	Ms.G.Chandrapraba
CP5152	Advanced Computer Architecture	Ms.P.Nalayini
CP5153	Operating System Internals	Mr.M.Arun
CP5154	Advanced Software Engineering	Ms.S.Puvaneswari
CP5191	Machine Learning Techniques	Ms.K.Abhirami
CP5161	Data Structures Laboratory	Mr.R.Sriramkumar

PG - II YEAR

SUB CODE	SUBJECT NAME	STAFF NAME
CP5005	Software Quality Assurance & Testing	Mr.M.Arun
CP5074	Social Network Analysis	Ms.Jasmine Smilee
CP5010	Reconfigurable Computing	Ms.P.Nalayini
CP5311	Project Work Phase I	Dr.S.M.Uma


HOD/CSE

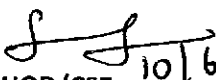
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

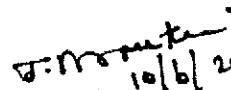
ACADEMIC YEAR 2019 – 2020 (ODD SEMESTER)

STAFF WORKLOAD

SNO	STAFF NAME	THEORY / PRACTICAL WITH SUB.CODE	CLASS / BRANCH WITH CLASS STRENGTH
1.	Dr.S.M.Uma HOD/CSE	T1: EC8691 -Microprocessor & Microcontroller T2: CP5152 – Advanced Computer Architecture L1: PG Project Phase I	T1: III CSE (47) T2: I ME CSE L1: II ME CSE (2)
2.	Ms.K.Abhirami (HEAD IQAC)	T1: Data Structures T2: CP5191 – Machine Learning Techniques L1: UG Project Phase I	T1: II CSE (44) T2: I ME CSE L1: IV CSE (40)
3.	Ms.S.Puvaneswari	T1: Theory of Computation T2: CP5154 – Advanced Software Engineering L1: Microprocessor & Microcontroller Lab L2: GE8161 – PSPP Lab	T1: III CSE (47) T2: II ME CSE (2) L1: III CSE (47) L2: I Year
4.	Ms.B.Sangeetha	T1: CS6703 – Grid and Cloud Computing T2: CS8392 – OOP L1: CS8383 – OOP Lab L2: GE8161 – PSPP Lab	T1: IV CSE (40) T2: III EEE (15) L1: III EEE (15) L2: I Year
5.	Mr.K.Rajesh	T1: Computer Networks	T1: III CSE (47)
6.	Ms.R.Ranitha	T1: Object Oriented Analysis & Design T2: GE8151 – Problem Solving & Python Programming L1: OOAD Lab L2: GE8161 – PSPP Lab	T1: III CSE (47) T2: I Year L1: III CSE (47) L2: I Year
7.	Mr.S.Rajarajan	T1: CS8351 - Digital Principles & System Design T2: CP5153 – Operating System Internals L1: CS8381 - Data Structures Lab	T1: II CSE (44) T2: I ME CSE L1: II CSE (44)
8.	Dr.D.Sivakumar	T1: CS6701 – Cryptography & Network Security T2: CS8392 – OOP L1: CS8393 – OOP Lab	T1: IV CSE(40) T2: II CSE (44) L1: II CSE (44)
9.	Ms.P.Nalayini	T1: Information Retrieval T2: CP5010 – Reconfigurable Computing T3: I Year ME CSE L1: Security Lab	T1: IV CSE (40) T2: II ME CSE (2) T3: I ME CSE L1: IV CSE (40)

SNO	STAFF NAME	THEORY / PRACTICAL WITH SUB.CODE	CLASS / BRANCH WITH CLASS STRENGTH
10.	Ms.R.Sugantha Lakshmi	T1: Open Elective – I T2: GE8151 – Problem solving & Python Programming L1: Networks Lab L2: GE8161 – PSPP Lab	T1: III CSE (47) T2: I Year L1: III CSE (47) L2: I Year
11.	Mr.R.Sriramkumar	T1: Cyber Forensics T2: GE8151 – Problem Solving & Python Programming L1: CS6712 – GCC Lab L2: CP5161 – Data Structures Lab	T1: IV CSE (40) T2: I Year L1: IV CSE(40) L2: ME I CSE
12.	Mr.M.Arun	T1: GE8151 Problem Solving & Python Programming T2: GE8151 Problem Solving & Python Programming T3: CP5005 – Software Quality Assurance L1: GE8161 – PSPP Lab	T1: I Year T2: I Year T3: II ME CSE L1: I Year
13.	Ms.G.Chandra Praba	T1: Data Structures T3: CP5151 – Advanced Data Structures & Algorithm L1: Data Structures Lab L2: Networks Lab	T1: II ECE T2: I Year T3: I ME CSE L1: II ECE L2: III ECE
14.	Ms.S.Jasmine Smilee	T1: CP5074 – Social Network Analysis	T1: II ME CSE


HOD/CSE 10/6/19


10/6/2019
PRINCIPAL



KINGS
COLLEGE OF ENGINEERING
(AAC Accredited Institution)
(Approved by AICTE, New Delhi, Affiliated to Anna University, Chennai)



Status of syllabus coverage as on - 19.07.19

Department		CSE				Year		II		Sec	A
Class Co-ordinator		Mr.S.Rajarajan				Mentor	Dr.D.Sivakumar		Ms.K.Abhirami		
Sl.No.	Name of the Subject / Code	Staff Name	Units Covered	Hours planned	Hours taken	No. of Hrs Req. to complete remaining portions	Reasons for deviation	Verification of student notes	Signature of faculty		
1.	MA8351/Discrete Mathematics	Ms.S.Geetha	1 1/2	19	22	-	-	Yes	19.7.19		
2.	CS8351/Digital Principles and System Design	Mr.S.Rajarajan	1 1/2	21	21	-	-	Yes	[Signature]		
3.	CS8391/Data Structures	Ms.K.Abhirami	1 3/4	19	19	-	-	Yes	[Signature]		
4.	CS8392/Object Oriented Programming	Dr.D.Sivakumar	1 1/2	18	18	-	-	Yes	[Signature]		
5.	EC8395/Communication Engineering	Mr.T.Jeyaseelan	2	20	23	-	-	Yes	[Signature]		
Sl.No.	Name of the Lab / Code	Staff Name	Experiment completed	Hours planned	Hours taken	No. of Hrs Req. to complete remaining portions	Reasons for deviation	Verification of observation / Records	Signature of faculty		
1.	CS8381/Data Structures Laboratory	Mr.S.Rajarajan & Ms.K.Abhirami	3b 3c	B1-16 B2-20	B1-15 B2-19	-	-	Yes	[Signature]		
2.	CS8382/Digital Systems Laboratory	Mr.R.Sathyaraj / Mr. N. Newton David Raj	3	B1-16 B2-16	B1-15 B2-15	-	-	Yes	[Signature]		
3.	CS8383/Object Oriented Programming Laboratory	Dr.D.Sivakumar & B.Sangeetha	4	B1-20 B2-16	B1-19 B2-15	-	-	Yes	[Signature]		
4.	HS8381/Interpersonal Skills/ Listening & Speaking	Mr.P.Rajeshwaran / Mrs. C. Jansi Rani	4	8	8	-	-	Yes	[Signature]		

Class Co-ordinator

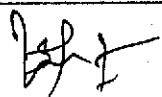
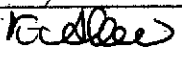
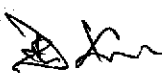
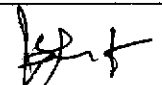
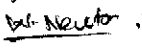
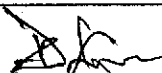
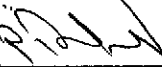
8
HOD 19.7.19



KINGS
COLLEGE OF ENGINEERING
(NAAC Accredited Institution)
(Approved by AICTE, New Delhi, Affiliated to
Anna University, Chennai)



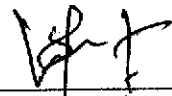
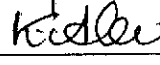
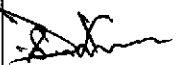
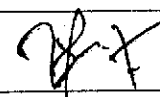
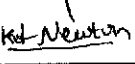
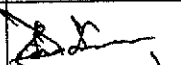
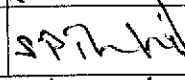
Status of syllabus coverage as on - 16.08.19

Department		CSE				Year		II		Sec	A
Class Co-ordinator		Mr.S.Rajarajan				Mentor		Dr.D.Sivakumar		Ms.K.Abhirami	
Sl.No.	Name of the Subject / Code	Staff Name	Units Covered	Hours planned	Hours taken	No. of Hrs Req. to complete remaining portions	Reasons for deviation	Verification of student notes	Signature of faculty		
1.	MA8351/Discrete Mathematics	Ms.S.Geetha	2.9	38	41	-	-	Yes			
2.	CS8351/Digital Principles and System Design	Mr.S.Rajarajan	3	44	41	-	orientation, Guest Lecture Assessment Test	-			
3.	CS8391/Data Structures	Ms.K.Abhirami	3.5	40	40	-	Algorithms based subject	-			
4.	CS8392/Object Oriented Programming	Dr.D.Sivakumar	3 1/4	40	36	-	Assignment Test	Yes			
5.	EC8395/Communication Engineering	Mr.T.Jeyaseelan	4.1	43	38	-	Yoga Function Guest Lecture Ass-I, class test	Yes			
Sl.No.	Name of the Lab / Code	Staff Name	Experiment completed	Hours planned	Hours taken	No. of Hrs Req. to complete remaining portions	Reasons for deviation	Verification of observation / Records	Signature of faculty		
1.	CS8381/Data Structures Laboratory	Mr.S.Rajarajan & Ms.K.Abhirami	8 8	32 36	30 34	-	-	Yes			
2.	CS8382/Digital Systems Laboratory	Mr.R.Sathiyaraj / Mr.W.Newton David Raj	RI - 8 BJ - 8	32 32	30 30	-	-	Yes			
3.	CS8383/Object Oriented Programming Laboratory	Dr.D.Sivakumar & B.Sangeetha	BI - 8 BJ - 7	32 28	32 27	-	-	Yes			
4.	HS8381/Interpersonal Skills/ Listening & Speaking	Mr.P.Rajeshwaran	9	18	18	-	-	Yes			

Class Co-ordinator

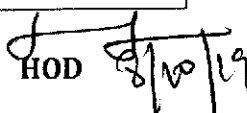
HOD 17/8/19

Status of syllabus coverage as on - 08.10.19

Department		CSE				Year		II		Sec	A
Class Co-ordinator		Mr.S.Rajarajan				Mentor		Dr.D.Sivakumar		Ms.K.Abhirami	
Sl.No.	Name of the Subject / Code	Staff Name	Units Covered	Hours planned	Hours taken	No. of Hrs Req. to complete remaining portions	Reasons for deviation	Verification of student notes	Signature of faculty		
1.	MA8351/Discrete Mathematics	Ms.S.Geetha	5	65	65	-	-	yes			
2.	CS8351/Digital Principles and System Design	Mr.S.Rajarajan	5	65	65	-	-	yes			
3.	CS8391/Data Structures	Ms.K.Abhirami	5	55	56	-	-	yes			
4.	CS8392/Object Oriented Programming	Dr.D.Sivakumar	5	50	50	-	-	yes			
5.	EC8395/Communication Engineering	Mr.T.Jeyaseelan	5	50	51	-	-	yes			
Sl.No.	Name of the Lab / Code	Staff Name	Experiment completed	Hours planned	Hours taken	No. of Hrs Req. to complete remaining portions	Reasons for deviation	Verification of observation / Records	Signature of faculty		
1.	CS8381/Data Structures Laboratory	Mr.S.Rajarajan & Ms.K.Abhirami	B1-12 B2-12	60	60	-	-	yes			
2.	CS8382/Digital Systems Laboratory	Mr.R.Sathyaraj / Mr. W. Newton David Raj	B1-10 B2-10	60	60	-	-	yes			
3.	CS8383/Object Oriented Programming Laboratory	Dr.D.Sivakumar & B.Sangeetha	B1-13 B2-13	60	62	-	-	yes			
4.	HS8381/Interpersonal Skills/ Listening & Speaking	Mr.P.Rajeshwaran	B1-13 B2-13	30	30	-	-	yes			

Class Co-ordinator

HOD


8/10/19

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Weekly Test I

CUMULATIVE MARK DETAILS - 2019-2020

Branch :Computer Science and Engineering

Year/Sem/Sec :II - Year - III SEM - A

			Date of Test	12/07/2019	08/07/2019	09/07/2019	10/07/2019	11/07/2019		
			Date of Submission	15/07/2019	11/07/2019	13/07/2019	13/07/2019	15/07/2019		
				EC8395-CE	MA8351-DM	CS8351-DPSD	CS8391-DS	CS8392-OOP		
SNO	Student Id	Student Name	Section	CE	DM	DPSD	DS	OOP	Total	Result
1	K18CS001	ABARNA M	A	18	16	13	19	17	83	PASS
2	K18CS002	ABINAYA N	A	17	18	17	20	10	82	PASS
3	K18CS003	ABIRAMI K	A	18	20	17	19	19	93	PASS
4	K18CS004	ABIRAMI M	A	19	16	13	17	16	81	PASS
5	K18CS005	ABIRAMI P	A	18	17	20	20	18	93	PASS
6	K18CS006	ABURVANAYAKI T	A	19	17	17	11	19	83	PASS
7	K18CS007	AISWARYA K	A	17	19	11	20	16	83	PASS
8	K18CS008	ARCHANA A	A	17	19	15	19	18	88	PASS
9	K18CS009	ARUNKUMAR G	A	13	AB	20	16	11	60	FAIL
10	K18CS010	BABU M	A	16	14	16	6	14	66	FAIL
11	K18CS011	DEEPIKA V	A	AB	OD	OD	OD	OD	0	FAIL
12	K18CS012	DHARSHINI T R	A	17	19	17	20	20	93	PASS
13	K18CS013	DIVYABHARATHI G	A	18	12	20	20	18	88	PASS
14	K18CS014	GAYATHRI J	A	17	20	20	16	19	92	PASS
15	K18CS015	GAYATHRI K	A	19	19	8	19	18	83	FAIL
16	K18CS016	GUNASEELAN B	A	12	AB	19	16	14	61	FAIL
17	K18CS017	HARIHARAN E	A	12	15	20	16	15	78	PASS
18	K18CS018	ISHWARYA S	A	18	18	16	19	17	88	PASS
19	K18CS019	JEMIMAESTHERGRACE M	A	AB	13	16	10	15	54	FAIL
20	K18CS020	KAMALA PRIYA M	A	18	16	17	20	19	90	PASS
21	K18CS021	KAVIYACEHLVAN K	A	13	15	16	8	15	67	FAIL
22	K18CS022	KEERTHANA V	A	18	17	19	18	15	87	PASS
23	K18CS024	NANDHA KISHORE S	A	5	10	10	12	12	49	FAIL
24	K18CS025	NASRIN BANU N	A	19	18	19	20	17	93	PASS
25	K18CS026	NATHIYADEVI N	A	17	13	18	10	15	73	PASS
26	K18CS027	NAVEEN SUNDHAR V	A	10	8	6	6	4	34	FAIL
27	K18CS028	PATHMANABAN R	A	13	19	19	10	16	77	PASS
28	K18CS029	PRIYA G	A	13	7	AB	7	11	38	FAIL
29	K18CS030	PRIYADHARSHINI M	A	14	17	AB	AB	16	47	FAIL
30	K18CS031	PRIYADHARSHINI R	A	18	16	17	20	17	88	PASS
31	K18CS032	RAMYA J	A	15	11	18	11	14	69	PASS
32	K18CS033	RATHNAKUMAR T	A	13	9	18	15	9	64	FAIL
33	K18CS034	SARATHKUMAR R	A	10	19	18	16	11	74	PASS
34	K18CS035	SELVARANI T	A	18	19	15	AB	20	72	FAIL
35	K18CS036	SIVARANJANI T	A	10	OD	OD	OD	OD	10	FAIL
36	K18CS037	SNEHA K	A	15	13	15	12	14	69	PASS
37	K18CS038	SNEKA S	A	19	19	13	20	16	87	PASS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Assessment Test I

CUMULATIVE MARK DETAILS - 2019-2020

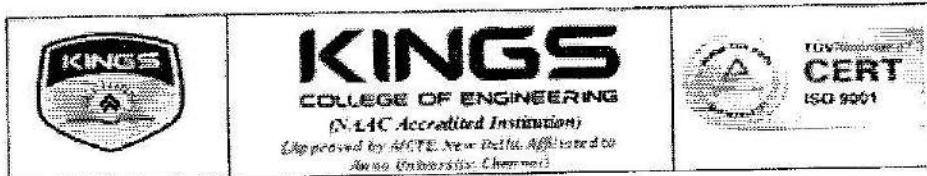
Branch :Computer Science and Engineering

Year/Sem/Sec :II - Year - III SEM - A

			Date of Test	26/07/2019	22/07/2019	23/07/2019	24/07/2019	25/07/2019		
			Date of Submission	29/07/2019	24/07/2019	26/07/2019	27/07/2019	29/07/2019		
				EC8395-CE	MA8351-DM	CS8351-DPSD	CS8391-DS	CS8392-OOP		
SNO	Student Id	Student Name	Section	CE	DM	DPSD	DS	OOP	Total	Result
1	K18CS001	ABARNA M	A	36	37	31	31	39	174	PASS
2	K18CS002	ABINAYA N	A	43	41	32	30	22	168	FAIL
3	K18CS003	ABIRAMI K	A	43	49	44	49	45	230	PASS
4	K18CS004	ABIRAMI M	A	31	30	24	37	31	153	FAIL
5	K18CS005	ABIRAMI P	A	43	47	38	46	40	214	PASS
6	K18CS006	ABURVANAYAKI T	A	45	48	37	42	36	208	PASS
7	K18CS007	AISWARYA K	A	40	30	15	48	38	171	FAIL
8	K18CS008	ARCHANA A	A	36	33	30	32	31	162	PASS
9	K18CS009	ARUNKUMAR G	A	41	16	30	38	30	155	FAIL
10	K18CS010	BABU M	A	25	18	37	30	17	127	FAIL
11	K18CS011	DEEPIKA V	A	30	33	20	22	31	136	FAIL
12	K18CS012	DHARSHINI T R	A	34	31	30	35	32	162	PASS
13	K18CS013	DIVYABHARATHI G	A	45	40	30	42	39	196	PASS
14	K18CS014	GAYATHRI J	A	37	47	38	49	49	220	PASS
15	K18CS015	GAYATHRI K	A	42	46	39	41	39	207	PASS
16	K18CS016	GUNASEELAN B	A	36	7	28	44	35	150	FAIL
17	K18CS017	HARIHARAN E	A	38	39	37	47	37	198	PASS
18	K18CS018	ISHWARYA S	A	44	45	34	46	40	209	PASS
19	K18CS019	JEMIMA ESTHER GRACE M	A	21	25	23	19	14	102	FAIL
20	K18CS020	KAMALA PRIYA M	A	34	41	31	27	37	170	FAIL
21	K18CS021	KAVIYACEHLVAN K	A	20	8	41	27	20	116	FAIL
22	K18CS022	KEERTHANA V	A	39	44	37	36	22	178	FAIL
23	K18CS024	NANDHA KISHORE S	A	28	25	21	18	24	116	FAIL
24	K18CS025	NASRIN BANU N	A	38	34	39	24	25	160	FAIL
25	K18CS026	NATHIYADEVI N	A	35	35	35	38	37	180	PASS
26	K18CS027	NAVEEN SUNDHAR V	A	16	5	32	7	6	66	FAIL
27	K18CS028	PATHMANABAN R	A	33	22	41	36	19	151	FAIL
28	K18CS029	PRIYA G	A	14	24	28	23	AB	89	FAIL
29	K18CS030	PRIYADHARSHINI M	A	24	22	14	24	18	102	FAIL
30	K18CS031	PRIYADHARSHINI R	A	43	43	43	48	31	208	PASS
31	K18CS032	RAMYA J	A	35	30	30	19	17	131	FAIL
32	K18CS033	RATHNAKUMAR T	A	32	18	41	32	26	149	FAIL
33	K18CS034	SARATHKUMAR R	A	30	18	38	31	30	147	FAIL
34	K18CS035	SELVARANI T	A	35	41	35	42	31	184	PASS
35	K18CS036	SIVARANJANI T	A	23	23	26	29	25	126	FAIL
36	K18CS037	SNEHA K	A	33	AB	17	18	19	87	FAIL
37	K18CS038	SNEKA S	A	42	23	30	31	36	162	FAIL

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Weekly Test II
CUMULATIVE MARK DETAILS - 2019-2020
Branch :Computer Science and Engineering
Year/Sem/Sec :II - Year - III SEM

			Date of Test	09/08/2019	05/08/2019	06/08/2019	07/08/2019	08/08/2019		
			Date of Submission	13/08/2019	07/08/2019	09/08/2019	13/08/2019	13/08/2019		
				EC8395-CE	MA8351-DM	CS8351-DPSD	CS8391-DS	CS8392-OOP		
SNO	Student Id	Student Name	Section	CE	DM	DPSD	DS	OOP	Total	Result
1	K18CS001	ABARNA M	A	19	18	18	20	18	91	PASS
2	K18CS002	ABINAYA N	A	18	18	18	20	17	88	PASS
3	K18CS003	ABIRAMI K	A	19	18	18	20	19	94	PASS
4	K18CS004	ABIRAMI M	A	18	17	18	20	17	79	PASS
5	K18CS005	ABIRAMI P	A	19	18	18	20	19	94	PASS
6	K18CS006	ABURVANAYAKI T	A	18	18	18	20	19	91	PASS
7	K18CS007	AISWARYA K	A	19	18	18	20	19	92	PASS
8	K18CS008	ARCHANA A	A	18	18	18	20	17	89	PASS
9	K18CS009	ARUNKUMAR G	A	19	18	17	20	19	93	PASS
10	K18CS010	BABU M	A	18	17	18	19	17	73	PASS
11	K18CS011	DEEPIKA V	A	19	18	18	20	17	90	PASS
12	K18CS012	DHARSHINI T R	A	18	18	18	20	18	92	PASS
13	K18CS013	DIVYABHARATHI G	A	18	18	18	20	19	93	PASS
14	K18CS014	GAYATHRI J	A	19	18	18	20	19	94	PASS
15	K18CS015	GAYATHRI K	A	19	18	18	20	17	90	PASS
16	K18CS016	GUNASEELAN B	A	19	18	17	20	19	93	PASS
17	K18CS017	HARIHARAN E	A	19	18	18	20	17	92	PASS
18	K18CS018	ISHWARYA S	A	19	18	17	20	18	92	PASS
19	K18CS019	JEMIMA ESTHER GRACE M	A	18	18	18	19	17	87	PASS
20	K18CS020	KAMALA PRIYA M	A	19	19	18	20	17	93	PASS
21	K18CS021	KAVIYACEHLVAN K	A	18	17	17	16	17	70	PASS
22	K18CS022	KEERTHANA V	A	19	18	18	20	18	91	PASS
23	K18CS024	NANDHA KISHORE S	A	17	17	18	AB	17	56	FAIL
24	K18CS025	NASRIN BANU N	A	19	18	18	20	17	89	PASS
25	K18CS026	NATHIYADEVI N	A	19	17	18	20	18	90	PASS
26	K18CS027	NAVEEN SUNDHAR V	A	17	17	18	16	17	53	PASS
27	K18CS028	PATHMANABAN R	A	18	17	18	20	17	82	PASS
28	K18CS029	PRIYA G	A	17	18	18	19	17	84	PASS
29	K18CS030	PRIYADHARSHINI M	A	18	18	17	20	17	86	PASS
30	K18CS031	PRIYADHARSHINI R	A	19	18	17	20	17	91	PASS
31	K18CS032	RAMYA J	A	19	18	18	17	19	75	PASS
32	K18CS033	RATHNAKUMAR T	A	18	18	18	20	19	84	PASS
33	K18CS034	SARATHKUMAR R	A	18	17	18	19	18	65	PASS
34	K18CS035	SELVARANI T	A	19	18	18	20	19	92	PASS
35	K18CS036	SIVARANJANI T	A	19	18	18	20	17	89	PASS
36	K18CS037	SNEHA K	A	19	17	18	17	18	62	PASS
37	K18CS038	SNFKA S	A	19	18	17	7	19	76	FAIL



Department of Computer Science and Engineering

ACADEMIC YEAR 2019-2020 (III SEM)

Assessment Test II EXAM - 06/09/2019

II - Year CSE 'A' [CS8391 - DATA STRUCTURES]

Conducted Period : 26

Roll No.	Name of the Student	Attn Period	Attn %	Attn Mark (10)	Assign Mark (20)	WT Mark Out of 20	AT Mark Out of 80	Tot Mark Out of 100
1	ABARNA M	26	100	10	20	20	56	76
2	ABINAYA N	25	96	10	20	20	40	60
3	ABIRAMI K	26	100	10	20	20	44	64
4	ABIRAMI M	26	100	10	20	20		56
5	ABIRAMI P	26	100	10	20	20	50	70
6	ABURVANAYAKI T	25	96	10	20	20	44	64
7	AISWARYA K	25	96	10	20	20	56	76
8	ARCHANA A	26	100	10	20	20	48	68
9	ARUNKUMAR G	21	81	8	20	20	40	60
10	BABU M	25	96	10	20	19		52
11	DEEPIKA V	26	100	10	20	20		54
12	DHARSHINI T R	25	96	10	20	20	40	60
13	DIVYABHARATHI G	26	100	10	20	20	52	72
14	GAYATHRI I	26	100	10	20	20	44	64
15	GAYATHRI K	26	100	10	20	20	40	60
16	GUNASEELAN B	24	92	9	20	20	48	68
17	HARIHARAN E	23	88	9	20	20	44	64
18	ISHWARYA S	23	88	9	20	20	46	66
19	JEMIMA ESTHER GRACE M	23	88	9	20	19		
20	KAMALA PRIYA M	26	100	10	20	20	44	64
21	KAVIYACEHLVAN K	26	100	10	20	16		55
22	KEERTHANA V	26	100	10	20	20	49	69
23	NANDHA KISHORE S	23	88	9	20			
24	NASRIN BANU N	26	100	10	20	20	40	60
25	NATHIYADEVI N	25	96	10	20	20	54	74

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MODEL EXAM

CUMULATIVE MARK DETAILS - 2019-2020

Branch :Computer Science and Engineering

Year/Sem/Sec :II - Year - III SEM

			Date of Test	15/10/2019	10/10/2019	11/10/2019	12/10/2019	14/10/2019		
			Date of Submission	16/10/2019	11/10/2019	12/10/2019	14/10/2019	15/10/2019		
				EC8395-CE	MA8351-DM	CS8351-DPSD	CS8391-DS	CS8392-OOP		
SNO	Student Id	Student Name	Section	CE	DM	DPSD	DS	OOP	Total	Result
1	K18CS001	ABARNA M	A	55	63	42	52	78	290	FAIL
2	K18CS002	ABINAYA N	A	52	72	36	41	79	280	FAIL
3	K18CS003	ABIRAMI K	A	80	93	70	78	91	412	PASS
4	K18CS004	ABIRAMI M	A	50	69	61	54	84	318	PASS
5	K18CS005	ABIRAMI P	A	80	93	63	83	85	404	PASS
6	K18CS006	ABURVANAYAKI T	A	60	70	68	72	82	352	PASS
7	K18CS007	AISWARYA K	A	52	78	85	65	79	359	PASS
8	K18CS008	ARCHANA A	A	50	41	42	41	54	228	FAIL
9	K18CS009	ARUNKUMAR G	A	55	70	50	63	50	288	PASS
10	K18CS010	BABU M	A	AB	50	51	44	58	203	FAIL
11	K18CS011	DEEPIKA V	A	53	63	34	39	59	248	FAIL
12	K18CS012	DHARSHINI T R	A	51	70	53	50	72	296	PASS
13	K18CS013	DIVYABHARATHI G	A	75	89	84	75	84	407	PASS
14	K18CS014	GAYATHRI J	A	70	73	67	66	78	354	PASS
15	K18CS015	GAYATHRI K	A	60	82	61	62	82	347	PASS
16	K18CS016	GUNASEELAN B	A	36	66	65	50	69	286	FAIL
17	K18CS017	HARIHARAN E	A	37	59	64	63	53	276	FAIL
18	K18CS018	ISHWARYA S	A	56	61	71	46	67	301	FAIL
19	K18CS019	JEMIMA ESTHER GRACE M	A	32	51	57	27	32	199	FAIL
20	K18CS020	KAMALA PRIYA M	A	61	77	57	22	57	274	FAIL
21	K18CS021	KAVIYACEHLVAN K	A	36	29	43	14	20	142	FAIL
22	K18CS022	KEERTHANA V	A	72	78	72	51	69	342	PASS
23	K18CS024	NANDHA KISHORE S	A	16	27	30	11	19	103	FAIL
24	K18CS025	NASRIN BANU N	A	50	35	AB	AB	31	116	FAIL
25	K18CS026	NATHIYADEVI N	A	65	67	66	39	69	306	FAIL
26	K18CS027	NAVEEN SUNDHAR V	A	10	37	18	1	8	74	FAIL
27	K18CS028	PATHMANABAN R	A	50	64	65	41	50	270	FAIL
28	K18CS029	PRIYA G	A	32	AB	AB	AB	AB	32	FAIL
29	K18CS030	PRIYADHARSHINI M	A	30	55	62	32	56	235	FAIL
30	K18CS031	PRIYADHARSHINI R	A	50	63	70	53	81	317	PASS
31	K18CS032	RAMYA J	A	50	58	41	21	44	214	FAIL
32	K18CS033	RATHNAKUMAR T	A	40	52	39	33	50	214	FAIL
33	K18CS034	SARATHKUMAR R	A	42	65	43	32	39	221	FAIL
34	K18CS035	SELVARANI T	A	52	81	72	63	73	341	PASS
35	K18CS036	SIVARANJANI T	A	50	62	53	36	51	252	FAIL
36	K18CS037	SNEHA K	A	33	45	40	19	30	167	FAIL
37	K18CS038	SNEHA S	A	65	51	38	44	50	248	FAIL

KINGS COLLEGE OF ENGINEERING, PUNALKULAM

Branch : II CSE

Year / Semester : II / III

Date: 17.09.19

Consolidated Mark Statement: Week Test 1 / Assessment Test 1 / Week Test 2 / Assessment Test 2 / Model Exam

Roll No.	MA8351 4.09.19		CS8351 05.09.19		CS8391 06.09.19		CS8392 07.09.19		EC8395 09.09.19		Result
	DM		DPSD		DS		OOP		CE		
	80	100	80	100	80	100	80	100	80	100	
1.	40	50	28	36	56	71	64	80	48	61	FAIL
2.	45	57	30	38	40	50	48	61	43	54	FAIL
3.	67	84	68	86	44	55	65	82	65	82	PASS
4.	53	67	28	35	36	45	53	67	49	62	FAIL
5.	64	80	50	63	50	63	56	71	64	80	PASS
6.	64	81	44	55	44	56	65	82	60	75	PASS
7.	56	70	44	56	56	70	64	80	64	80	PASS
8.	45	57	40	50	48	60	48	61	52	65	PASS
9.	44	56	40	50	40	50	59	74	52	65	PASS
10.	46	58	29	37	32	40	36	45	41	52	FAIL
11.	45	57	40	50	34	43	47	59	56	70	FAIL
12.	53	67	26	33	40	50	54	68	40	50	FAIL
13.	63	79	57	72	52	65	69	87	72	90	PASS
14.	64	81	50	63	44	55	71	89	64	80	PASS
15.	56	71	59	74	40	50	59	74	52	65	PASS
16.	63	79	48	61	48	60	63	79	40	50	PASS
17.	63	79	32	40	44	56	52	66	32	41	FAIL
18.	58	73	48	61	46	58	66	83	60	75	PASS
19.	46	58	40	50	26	33	36	46	40	50	FAIL
20.	64	80	48	60	44	55	60	76	56	70	PASS
21.	0	01	15	19	35	44	25	32	32	41	FAIL
22.	61	77	57	72	49	62	53	67	52	66	PASS
23.	44	56	32	40	26	33	18	23	20	26	FAIL
24.	49	62	41	52	40	50	48	60	52	65	PASS
25.	61	77	46	58	54	68	66	83	52	65	PASS
26.	12	16	8	11	16	20	7	09	16	20	FAIL
27.	61	77	41	52	43	54	46	58	51	64	PASS
28.	44	56	40	50	40	50	32	41	40	50	FAIL
29.	37	47	25	32	28	35	53	67	40	50	FAIL
30.	60	76	43	54	52	66	69	87	40	51	PASS
31.	8	10	23	29	22	28	28	36	40	50	FAIL
32.	54	68	33	42	40	51	50	63	36	45	FAIL
33.	60	76	40	50	35	44	43	54	44	56	FAIL
34.	56	70	52	66	56	70	69	87	51	64	PASS
35.	48	61	40	50	50	63	56	71	56	70	PASS
36.	34	43	20	25	28	36	47	59	40	50	FAIL
37.	32	41	36	45	35	44	61	77	52	65	PASS
38.	43	54	32	40	35	44	31	39	40	50	FAIL

Roll No.	MA8351 4.09.19 DM		CS8351 05.09.19 DPSD		CS8391 06.09.19 DS		CS8392 07.09.19 OOP		EC8395 09.09.19 CE		Result
	80	100	80	100	80	100	80	100	80	100	
39.	43	54	41	52	37	47	60	76	36	45	FAIL
40.	45	57	47	59	52	65	62	78	41	52	PASS
41.	36	45	40	51	34	43	55	69	24	31	FAIL
42.	OD	OD	29	37	40	51	52	66	48	60	FAIL
43.	49	62	41	52	30	38	62	78	AB	AB	FAIL
44.	7	09	1	02	7	09	10	13	4	05	FAIL
45.	47	59	26	33	18	23	25	32	36	45	FAIL

Total No. of Students:45

Details	MA8351 4.09.19 DM	CS8351 05.09.19 DPSD	CS8391 06.09.19 DS	CS8392 07.09.19 OOP	EC8395 09.09.19 CE
Absent	1	0	0	0	1
Appeared	44	45	45	45	44
Pass	36	27	27	35	35
Fail	8	18	18	10	9
Pass %	81.82	60	60	77.78	79.55

SUB CODE	COURSE_SHORTCODE	SUBJET NAME	STAFF NAME
EC8395	CE	COMMUNICATION ENGINEERING	T.Jeyaseelan
MA8351	DM	DISCRETE MATHEMATICS	Geetha S
CS8351	DPSD	DIGITAL PRINCIPLES AND SYSTEM DESIGN	Rajarajan.S
CS8391	DS	DATA STRUCTURES	Abhirami.K
CS8392	OOP	OBJECT ORIENTED PROGRAMMING	Sivakumar.D

OVERALL PASS %=21/45=46.4

Signature of the coordinator

Name: (S RAJARAJANI)

HOD / CSE

J. Rajarajani
13/9/2019



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ACADEMIC YEAR 2019 - 2020 / ODD SEMESTER

Guidelines for Remedial class

- It is planned to conduct the Phase 1 coaching class to support the students who scored less than 16 out of 20 in weektest-I during 4.40-5.30 pm. They will write retest during that time.
- Phase 2 coaching class will be conducted for the students those who secure less than 16 out of 20 in weektest-II during 4.40-5.30 pm and they have to write retest.
- Students those who secured Less than 70 marks in assessment II will attend the coaching classes on Saturdays. They have to learn and write test during coaching.
- Special coaching will be given to the students except all clear students during afternoon session of model examination.
- Intensive coaching class will be conducted from 21.10.19-1.11.19

[Handwritten Signature]
5/7/2019
H.O.O/LSC



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ACADEMIC YEAR 2019 - 20 ODD SEMESTER
SATURDAY SPECIAL TIME TABLE - 13.07.19

Session	1	2	10.45 am - 11.00 am	3	4	12.30 pm - 01.15 pm	5	6	02.45 pm - 03.00 pm	7	8
Day	09.15am - 10.00am	10.00am - 10.45am		11.00am - 11.45am	11.45am - 12.30pm		01.15pm - 02.00pm	02.00pm - 02.45pm		03.00pm - 03.45pm	03.45pm - 04.30pm
II CSE	CS8391	CS8392	BREAK	CS8351		LUNCH BREAK	EC8395		BREAK	MA8351	
III CSE	EC8691	OMF551		CS8591	OMF551		IIT BOMBAY TUTORIAL			MA8551	CS8592
IV CSE	CS6704	CS6703		CS6004	CS6701		CS6007	CS6702		IIT BOMBAY TUTORIAL	

SUB CODE	NAME OF THE SUBJECT	NAME OF THE STAFF
II CSE		
MA8351	Discrete Mathematics	Ms.S.Geetha
CS8351	Digital Principles and System Design	Mr.S.Rajarajan
CS8391	Data Structures	Ms.K.Abhirami
CS8392	Object Oriented Programming	Dr.D.Sivakumar
EC8395	Communication Engineering	Mr.T.Jeyaseelan
III CSE		
MA8551	Algebra and Number Theory	Mr.G.Jeyakrishnan
CS8591	Computer Networks	Mr.K.Rajesh
EC8691	Microprocessor & Microcontroller	Dr.S.M.Uma
CS8592	Object Oriented Analysis & Design	Ms.R.Ranitha
OMF551	Product Design and Development	Ms.R.Suganthalakshmi
	IIT BOMBAY TUTORIAL	Ms.K.Abhirami
IV CSE		
CS6701	Cryptography and Network Security	Dr.D.Sivakumar
CS6702	Graph Theory and Applications	Dr.R.Suresh
CS6703	Grid and Cloud Computing	Ms.B.Sangeetha
CS6704	Resource Management Techniques	Dr.G.Shankara Kalidoss
CS6004	Cyber Forensics	Mr.R.Sriramkumar
	IIT BOMBAY TUTORIAL	Mr.R.Sriramkumar

W. S. P.
CLASS INCHARGE

S. S.
HOD/CSE



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ACADEMIC YEAR 2019 – 2020 (ODD SEMESTER)
COACHING CLASS SCHEDULE

Year	Date	Sub Code	Subject Name	Staff Name
II	05.08.19	MA8351	Discrete Mathematics	Ms.S.Geetha
	06.08.19	CS8351	Digital Principles and System Design	Mr.S.Rajarajan
	07.08.19	CS8391	Data Structures	Ms.K.Abhirami
	08.08.19	CS8392	Object Oriented Programming	Dr.D.Sivakumar
	09.08.19	EC8395	Communication Engineering	Mr.T.Jeyaseelan
III	05.08.19	MA8551	Algebra and Number Theory	Mr.G.Jeyakrishnan
	06.08.19	CS8591	Computer Networks	Mr.K.Rajesh
	07.08.19	EC8691	Microprocessor & Microcontroller	Dr.S.M.Uma
	08.08.19	CS8501	Theory of Computation	Ms.S.Puvaneswari
	09.08.19	CS8592	Object Oriented Analysis & Design	Ms.R.Ranitha
	13.08.19	OMF551	Product Design and Development	Ms.R.Suganthalakshmi
IV	05.08.19	CS6701	Cryptography and Network Security	Dr.D.Sivakumar
	06.08.19	CS6702	Graph Theory and Applications	Dr.R.Suresh
	07.08.19	CS6703	Grid and Cloud Computing	Ms.B.Sangeetha
	08.08.19	CS6704	Resource Management Techniques	Dr.G.Shankara Kalidoss
	09.08.19	CS6004	Cyber Forensics	Mr.R.Sriramkumar
	13.08.19	CS6007	Information Retrieval	Ms.P.Nalayini

[Signature]
HOD/CSE



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ACADEMIC YEAR 2019 – 2020 (ODD SEMESTER)
ASSESSMENT TEST – II REVISION CLASS SCHEDULE

Year	Date	Sub Code	Subject Name	Staff Name
II	03.09.19	MA8351	Discrete Mathematics	Ms.S.Geetha & Mr.S.Rajaraman
	04.09.19	CS8351	Digital Principles and System Design	Mr.S.Rajaraman & Ms.K.Abhirami
	05.09.19	CS8391	Data Structures	Ms.K.Abhirami & Mr.S.Rajaraman
	06.09.19	CS8392	Object Oriented Programming	Dr.D.Sivakumar & Mr.T.Jeyaseelan
	07.09.19	EC8395	Communication Engineering	Mr.T.Jeyaseelan & Dr.D.Sivakumar
III	04.09.19	CS8591	Computer Networks	Mr.K.Rajesh & Ms.R.Ranitha
	05.09.19	EC8691	Microprocessor & Microcontroller	Dr.S.M.Uma & Mr.K.Rajesh
	06.09.19	CS8501	Theory of Computation	Ms.S.Puvaneswari & Ms.R.Suganthalakshmi
	07.09.19	CS8592	Object Oriented Analysis & Design	Ms.R.Ranitha & Ms.S.Puvaneswari
	09.09.19	OMF551	Product Design and Development	Ms.R.Suganthalakshmi & Ms.S.Puvaneswari
IV	03.09.19	CS6701	Cryptography and Network Security	Dr.D.Sivakumar & Mr.R.Sriramkumar
	04.09.19	CS6702	Graph Theory and Applications	Dr.R.Suresh & Dr.G.Shankara Kalidoss
	05.09.19	CS6703	Grid and Cloud Computing	Ms.B.Sangeetha & Ms.P.Nalayini
	06.09.19	CS6704	Resource Management Techniques	Dr.G.Shankara Kalidoss & Dr.R.Suresh
	07.09.19	CS6004	Cyber Forensics	Mr.R.Sriramkumar & Dr.D.Sivakumar
	09.09.19	CS6007	Information Retrieval	Ms.P.Nalayini & Ms.B.Sangeetha

II CSE –

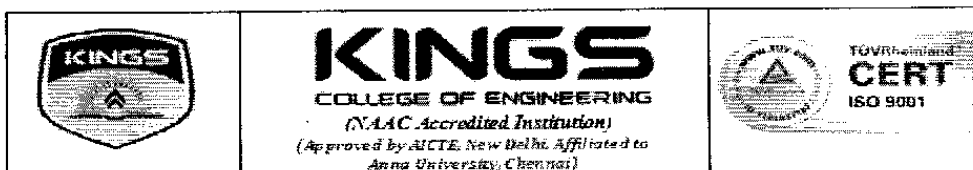
III CSE –

IV CSE –

CLASS INCHARGE

HOD/CSE

[Signature] 24/9/19



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ACADEMIC YEAR 2019 - 20 ODD SEMESTER
SATURDAY COACHING CLASS TIME TABLE

Session	DATE	1	2	10.45 am	3	4	12.30 pm	5	6	02.45 pm	7	8
Day		09.15am - 10.00am	10.00am - 10.45am	- 11.00 am	11.00am - 11.45am	11.45am - 12.30pm	- 01.15 pm	01.15pm - 02.00pm	02.00pm - 02.45pm	- 03.00 pm	03.00pm - 03.45pm	03.45pm - 04.30pm
I CSE	14.09.19	CS8391	CS8351	BREAK	CS8392		LUNCH BREAK	EC8395		BREAK	MA8351	
	28.09.19	CS8391			CS8351			EC8395			CS8392	
II CSE	14.09.19	CS8501			CS8501	EC8691		EC8691			MA8551	
	28.09.19	CS8591			CS8591	OMF551		OMF551			CS8592	
IV CSE	14.09.19	CS6703			CS6007			CS6004			CS6701	
	28.09.19	CS6704			CS6701			CS6702			CS6703	CS6704

SUB CODE	NAME OF THE SUBJECT	NAME OF THE STAFF
II CSE		
MA8351	Discrete Mathematics	Ms.S.Geetha
CS8351	Digital Principles and System Design	Mr.S.Rajarajan
CS8391	Data Structures	Ms.K.Abhirami
CS8392	Object Oriented Programming	Dr.D.Sivakumar
EC8395	Communication Engineering	Mr.T.Jeyaseelan
III CSE		
MA8551	Algebra and Number Theory	Mr.G.Jeyakrishnan
CS8591	Computer Networks	Mr.K.Rajesh
EC8691	Microprocessor & Microcontroller	Dr.S.M.Uma
CS8592	Object Oriented Analysis & Design	Ms.R.Ranitha
OMF551	Product Design and Development	Ms.R.Suganthalakshmi
CS8501	Theory of Computation	Ms.S.Puvaneswari
IV CSE		
CS6701	Cryptography and Network Security	Dr.D.Sivakumar
CS6702	Graph Theory and Applications	Dr.R.Suresh
CS6703	Grid and Cloud Computing	Ms.B.Sangeetha
CS6704	Resource Management Techniques	Dr.G.Shankara Kalidoss
CS6004	Cyber Forensics	Mr.R.Sriramkumar
CS6007	Information Retrieval Techniques	Ms.P.Nalayini

W. Cse - S. Puv
CLASS INCHARGE

S. S. 12/9/19
HOD/CSE



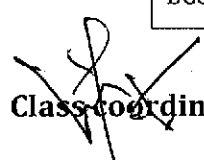
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ACADEMIC YEAR 2019-2020 (ODD SEMESTER)
ICC- TIME TABLE with effect from 16.10.2019

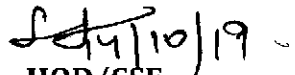
Year / Sem : II/ III

S.NO	DATE	SUBJECT CODE & NAME (FN)		SUBJECT CODE & NAME (AN)	
		9.15 a.m. to 12.30 p.m.		1. 15 p. m to 4.30p.m	
		Hall No.222(1-22)	Hall No. 227 (23-45)	Hall No.222(1-22)	Hall No. 227 (23-45)
1.	16.10.19 WEDNESDAY	MA8351 - Discrete Mathematics		CS8351 - Digital Principles and System Design	CS8391 - Data Structures
2.	17.10.19 THURSDAY	CS8351 - Digital Principles and System Design	CS8391 - Data Structures	MA8351 - Discrete Mathematics	
3.	18.10.19 FRIDAY	CS8392 - Object Oriented Programming	EC8395 - Communication Engineering	CS8391 - Data Structures	CS8351 - Digital Principles and System Design
4.	19.10.19 SATURDAY	MA8351 - Discrete Mathematics		EC8395 - Communication Engineering	CS8392 - Object Oriented Programming
5.	25.10.19 FRIDAY	CS8391 - Data Structures	CS8351 - Digital Principles and System Design	CS8392 - Object Oriented Programming	EC8395 - Communication Engineering
6	26.10.19 SATURDAY	EC8395 - Communication Engineering	CS8392 - Object Oriented Programming	MA8351 - Discrete Mathematics	
7	28.10.19 MONDAY	CS8391 - Data Structures	CS8351 - Digital Principles and System Design	CS8392 - Object Oriented Programming	EC8395 - Communication Engineering

S.NO	DATE	SUBJECT CODE & NAME (FN)		SUBJECT CODE & NAME (AN)	
		9.15 a.m. to 12.30 p.m.		1. 15 p. m to 4.30p.m	
		Hall No.222	Hall No. 227	Hall No.222	Hall No. 227
8	29.10.19 TUESDAY	MA8351 - Discrete Mathematics		EC8395 - Communication Engineering	CS8392 - Object Oriented Programming
9	30.10.19 WEDNESDAY	CS8391 - Data Structures	CS8351 - Digital Principles and System Design	CS8392 - Object Oriented Programming	EC8395 - Communication Engineering
10	31.10.19 THURSDAY	EC8395 - Communication Engineering	CS8392 - Object Oriented Programming	MA8351 - Discrete Mathematics	
11	01.11.19 FRIDAY	CS8351 - Digital Principles and System Design	CS8391 - Data Structures	MA8351 - Discrete Mathematics	

SUB CODE	NAME OF THE SUBJECT	NAME OF THE STAFF	DEPT
MA8351	Discrete Mathematics	Ms.S.Geetha	MATHS
CS8351	Digital Principles and System Design	Mr.S.Rajarajan	CSE
CS8391	Data Structures	Ms.K.Abhirami	CSE
CS8392	Object Oriented Programming	Dr.D.Sivakumar	CSE
EC8395	Communication Engineering	Mr.T.Jeyaseelan	ECE


Class Coordinator


HOD/CSE

Staff Signature



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ACADEMIC YEAR 2019 - 2020 / ODD SEMESTER

TEST REPORT													
Department		CSE							Year		II		
Name of the subject with Code		Data Structures - CS8391							Name of the Staff		K. ABHIRAMI		
Test	Date	No. of Students							Reason for poor performance	Corrective action	Signature of the staff	Signature of the HOD	Principal
		Total	Appeared	Absent	Passed	Pass %	60-80	81-100					
Class Test-1	10-7-19	45	38	7	29	75.3	-	-	Students need to practice sorting algorithm to perform various operations. Re test conducted. Additional practice test planned.	Kiddee 15/7/19	S. J. 15/9/19	S. J. 15/11/19	
Assessment Test -1	24-7-19	45	44	1	29	65.9	29	10	Few students didn't attend all questions. Application based questions. Periodically additional class tests were conducted. Students were given coaching.	Kiddee 29/7/19	S. J. 29/9/19	S. J. 29/11/19	
Class Test-2	7-8-19	45	41	4	40	88.89	-	-	Except absentee & failure candidate, others improved. (Better performance). Evening coaching classes, additional practice session, Lab slot helped in improvement.	Kiddee 12/8/19	S. J. 12/9/19	S. J. 12/11/19	
Assessment Test -2	6-9-19	45	45	-	27	60	10	17	Few students did not attend all questions. Management issue. (100 marks). Advised & guided students. More practice given to students.	Kiddee 11/9/19	S. J. 11/9/19	S. J. 11/9/19	
Model Exam	12-10-19	45	42	3	16	35.56	09	01	Lack of preparation for exam. Standard based questions were discussed & solved. Feedback given collected. Students were guided.	Kiddee 17/10	S. J. 17/10	S. J. 17/10/19	
AU Exam	18-11-19	45	45	-	35	77.77			Internal assessment. No case revised. Due feedback of preparation & regular. (Students advised). Students were guided & motivated for better performance.	Kiddee	S. J. 18/11/19	S. J. 18/11/19	

CT1, CT2
(out of 20)

Uty-Grades
A+ - 1
A - 2
B+ - 12
B - 20

Note: - Report should be retained by HOD concerned

Staff Timetable

Branch : CSE

Staff : ABHIRAMI.K

DAY	1	2	3	4	5	6	7	8
Monday	CSE II - Year III SEM A CS8391- DS							
Tuesday					CSE II - Year III SEM A CS8391- DS			
Wednesday					CSE II - Year III SEM A CS8381-DSLAB			
Thursday		CSE II - Year III SEM A CS8391- DS		CSE IV - Year VII SEM A PROJ2013- CDC PROJ	CSE II - Year III SEM A CS8381-DSLAB			
Friday							CSE II - Year III SEM A CS8391- DS	
Saturday								
Time	9:15 AM - 10:00 AM	10:00 AM - 10:45 AM	11:00 AM - 11:45 AM	11:45 AM - 12:30 PM	1:15 PM - 2:00 PM	2:00 PM - 2:45 PM	3:00 PM - 3:45 PM	3:45 PM - 4:30 PM

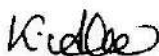
Course Code	Course Name	Tutorial Type	Section
CS8391	DATA STRUCTURES	Theoretical	CSE II - Year III SEM A
CS8381	DATA STRUCTURES LABORATORY	Practical	CSE II - Year III SEM A
PROJ2013	Project Work	Theoretical	CSE IV - Year VII SEM A

No. of Intervals :

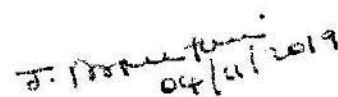
MORNING BREAK: 10:45 AM - 11:00 AM

LUNCH BREAK: 12:30 PM - 1:15 PM

EVENING BREAK: 2:45 PM - 3:00 PM


 STAFF


 HOD


 04/11/2019
 PRINCIPAL



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ACADEMIC YEAR 2019-20 / ODD SEMESTER**

II CSE NAME LIST

S.NO	REGISTER NO	NAME
1.	821118104001	Abarna M
2.	821118104002	Abinaya N
3.	821118104003	Abirami K
4.	821118104004	Abirami M
5.	821118104005	Abirami P
6.	821118104006	Aburvanayaki T
7.	821118104007	Aiswarya K
8.	821118104008	Archana A
9.	821118104009	Arunkumar G
10.	821118104010	Babu M
11.	821118104011	Deepika V
12.	821118104012	Dharshini T R
13.	821118104013	Divyabharathi G
14.	821118104014	Gayathri J
15.	821118104015	Gayathri K
16.	821118104016	Gunaseelan B
17.	821118104017	Hariharan E
18.	821118104018	Iswarya S
19.	821118104019	Jemima Esther Grace M
20.	821118104020	Kamalapriya M
21.	821118104021	Kaviyachelvan K
22.	821118104022	Keerthana V
23.	821118104024	Nandha kishore S
24.	821118104025	Narsrin banu N
25.	821118104026	Nathiya devi N
26.	821118104027	Naveen sundhar V
27.	821118104028	Pathmanaban R
28.	821118104029	Priya G
29.	821118104030	Priyadharshini M
30.	821118104031	Priyadharshini R
31.	821118104032	Ramya J
32.	821118104033	Rathnakumar T
33.	821118104034	Sarathkumar R
34.	821118104035	Selvarani T
35.	821118104036	Sivaranjani T
36.	821118104037	Sneha K
37.	821118104038	Sneka S
38.	821118104039	Sowmiya K
39.	821118104040	Srinithi C
40.	821118104041	Suruthi S
41.	821118104042	Surya R
42.	821118104043	Vaishnavi P
43.	821118104044	Vithyatharan S
44.		Gopinathan.D
45.		Sindu.P



KINGS

COLLEGE OF ENGINEERING
Punalkulam, Thanjavur.

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai



ISO 9001 : 2008

Attendance and Assessment Record

Name of the Staff : K-ABHIRAMI

Department : CSE

Subject Code & Name : CS8391, Data Structures

Branch : CSE

Semester : III

Attendance and Assessment Record

Name of the Staff : KABHIRAMI Dept CSE
 Name of the Subject : DATA STRUCTURES Code CS8391
 Branch CSE
 Semester III Year II
 Date of Commencement : 20.6.19 Last Working Day 19.10.19

Details	Sessions Planned	Sessions Handled	% of Portions covered	Sign. of HOD
End of the First Month	28	28	49/-	<i>S/S</i> 5/8
End of the Second Month	50	48	98/-	<i>S/S</i> 31/8
End of the Third Month	50 + Rev. slots	65	100/-	<i>S/S</i> 30/5
End of the Fourth Month	50 + Rev. + ICC classes	52 + 19 + 8 79* includes revision + ICC		<i>S/S</i> 2/11

J. Murugesu
04/11/2021

PRINCIPAL

Attendance Particulars												
Roll No.	Name	Date	21	24	25	26	27	28	1	2	4	5
		Month	6	6	6	6	6	7	7	7	7	7
		Period	7	1	5	0	2	7	1	6	1	7
1	Abarna M		/	/	/	/	/	/	/	/	/	/
2	Abinaya N		/	/	/	/	/	/	/	/	/	/
3	Abirami K		/	/	/	/	/	/	/	/	/	/
4	Abirami M		/	/	/	/	/	/	/	/	/	/
5	Abirami P		/	/	/	/	/	/	/	/	/	/
6	Abiramiyaki T		/	/	/	/	/	/	/	/	/	/
7	Aiswarya K		/	/	/	/	/	/	/	/	/	/
8	Aishwarya A		/	/	/	/	/	/	/	/	/	/
9	Arun Kumar G		/	/	/	/	/	/	/	/	/	/
10	Babu M		/	/	/	/	/	/	/	/	/	/
11	Deepika D		/	/	/	/	/	/	/	/	/	/
12	Dhruvshini T R		/	/	/	/	/	/	/	/	/	/
13	Divyabharathi G		/	/	/	/	/	/	/	/	/	/
14	Gayathri T		/	/	/	/	/	/	/	/	/	/
15	Gayathri K		/	/	/	/	/	/	/	/	/	/
16	Gunaseelan B		/	/	/	/	/	/	/	/	/	/
17	Harisharan E		/	/	/	/	/	/	/	/	/	/
18	Iswarya S		/	/	/	/	/	/	/	/	/	/
19	Jemima Esther Grace M		/	/	/	/	/	/	/	/	/	/
20	Kamalapranya M		/	/	/	/	/	/	/	/	/	/
21	Kavyachelvan K		/	/	/	/	/	/	/	/	/	/
22	Keerthana V		/	/	/	/	/	/	/	/	/	/
23	Nandha Kishore S		/	/	/	/	/	/	/	/	/	/
24	Nandini banu M		/	/	/	/	/	/	/	/	/	/
25	Natthya devi N		/	/	/	/	/	/	/	/	/	/

Roll No.	8	9	10	11	12	13	15	16	18	19
	7	7	7	7	7	7	7	7	7	7
	1	2	5	2	7	1	1	5	2	8
1	/	/	/	/	/	/	/	/	/	/
2	/	/	/	/	/	/	/	/	/	/
3	/	/	/	/	/	/	/	/	/	/
4	/	/	/	/	/	/	/	/	/	/
5	/	/	/	/	/	/	/	/	/	/
6	/	/	/	/	/	/	/	/	/	/
7	/	/	/	/	/	/	/	/	/	/
8	/	/	/	/	/	/	/	/	/	/
9	/	/	/	/	/	/	/	/	/	/
10	/	/	/	/	/	/	/	/	/	/
11	/	/	/	/	/	/	/	/	/	/
12	/	/	/	/	/	/	/	/	/	/
13	/	/	/	/	/	/	/	/	/	/
14	/	/	/	/	/	/	/	/	/	/
15	/	/	/	/	/	/	/	/	/	/
16	/	/	/	/	/	/	/	/	/	/
17	/	/	/	/	/	/	/	/	/	/
18	/	/	/	/	/	/	/	/	/	/
19	/	/	/	/	/	/	/	/	/	/
20	/	/	/	/	/	/	/	/	/	/
21	/	/	/	/	/	/	/	/	/	/
22	/	/	/	/	/	/	/	/	/	/
23	/	/	/	/	/	/	/	/	/	/
24	/	/	/	/	/	/	/	/	/	/
25	/	/	/	/	/	/	/	/	/	/

Attendance Particulars													
Roll No.	Name	Date	21	24	25	26	27	28	1	2	4	5	
		Month	6	6	6	6	6	6	7	7	7	7	
		Period	7	1	5	1	2	7	1	6	1	7	
26	Naveen Sondham V		1	a	1	a	a	1	1	1	1	1	
27	Pathmanaban R		1	1	1	1	1	1	1	1	1	1	
28	Priya G		a	a	a	a	a	a	1	1	1	a	
29	Priyadharsini M		a	a	1	1	1	1	a	1	1	1	
30	Priyadharsini R		1	1	1	1	1	1	1	1	1	1	
31	Ramya J		1	1	1	1	1	1	1	1	1	1	
32	Rathnakumar T		1	1	1	1	1	1	1	1	1	1	
33	Sarathkumar R		1	a	a	1	1	1	1	1	1	1	
34	Selvarani T		1	1	1	1	1	1	1	1	1	1	
35	Sivaranjani T		1	1	1	1	1	1	1	OD	OD	OD	
36	Sneha K		1	1	1	1	1	1	1	1	1	1	
37	Sneha S		1	1	1	1	1	1	1	1	1	1	
38	Sowmya K		1	1	1	1	1	1	1	OD	OD	OD	
39	Srinithi C		1	1	1	1	1	1	a	OD	OD	OD	
40	Suruthi S		1	1	1	1	1	1	1	1	1	1	
41	Surya R		a	1	1	1	1	1	a	a	1	a	
42	Varishnavi P		1	1	1	1	1	1	1	OD	OD	OD	
43	Vithyatharan S		1	1	1	1	1	1	1	1	1	1	
44	Vipinath D		a	1	1	1	1	1	a	1	1	a	
45	Sindhu R												
46													
47	No. of Student Absent		6	5	3	4	4	6	8	2	2	6	
48	No. of Student on OD		-	-	-	-	-	-	-	5	5	5	
49	No. of Student present		38	39	41	40	40	38	36	37	37	33	
50	Staff Sign		Kd	Kd	Kd	Kd	Kd	Kd	Kd	Kd	Kd	Kd	

Roll No.	8	9	10	11	12	13	15	16	18	19	
	7	7	7	7	7	7	7	7	7	7	
	1	2	5	2	7	1	1	5	2	2	
26	1	1	1	1	1	1	1	1	a	1	
27	1	1	1	1	1	1	1	1	1	OD	
28	1	a	1	1	1	1	1	1	1	1	
29	1	a	a	1	1	1	1	1	1	1	
30	1	1	1	1	1	1	a	1	1	1	
31	1	1	1	1	1	1	1	1	1	1	
32	1	1	1	1	1	1	1	1	1	OD	
33	1	1	1	1	1	1	1	1	1	1	
34	1	1	a	1	1	1	a	1	1	OD	
35	OD	OD	OD	OD	1	1	1	1	1	1	
36	1	1	1	1	1	a	a	1	a	1	
37	1	1	1	1	1	1	1	1	1	1	
38	OD	OD	OD	OD	a	1	1	1	1	1	
39	OD	OD	OD	OD	a	a	1	1	1	1	
40	1	1	1	1	1	1	1	1	1	a	
41	1	1	1	a	1	1	1	1	1	a	
42	OD	OD	OD	OD	a	1	1	1	1	1	
43	a	1	1	1	1	1	1	1	1	OD	
44	1	1	1	a	a	a	a	a	a	a	
45	1	a	1	1	1	1	1	1	1	1	
46											
47	3	3	2	2	6	6	5	2	4	5	
48	5	5	5	5	-	-	-	-	-	10	
49	38	39	38	38	39	39	40	43	41	30	
50	Kd	Kd	Kd	Kd	Kd	Kd	Kd	Kd	Kd	Kd	

Roll No.	Name	Date
		Month
		Period
1	Abarna M	
2	Abinaya N	
3	Abirami K	
4	Abirami M	
5	Abirami P	
6	Aburvarayaki T	
7	Aiswarya K	
8	Aishana U	
9	Arun Kumar G	
10	Babu M	
11	Deepika D	
12	Dharmshini T R	
13	Divya bhavathi G	
14	Gayathri J	
15	Gayathri K	
16	Gunaselvan B	
17	Haritharan E	
18	Iswarya S	
19	Jemima Esther Gra	
20	Kamalapritha M	
21	Kanvyachelvank K	
22	Keerthana V	
23	Nandha Kishore S	
24	Narsin banu M	
25	Nathaya devi N	

[illegible]

Assignment (Date)			Attendance				Test							
Roll No.	Announcement		Submission	Date				AT-I	AT-II	AT-III Max	CT-1	CT-2		Over all Attendance
1	20	20	16	19	16	16	31	74	52	19	20		3	100 Bt
2	20	20	16	18	16	16	30	50	41	20	20		2	98 B
3	20	20	15	19	16	16	49	55	78	19	20		3	98 A+
4	20	20	15	19	16	16	37	46	34	17	20		2	98 Bt
5	20	20	16	19	16	16	46	63	83	20	20		3	100 Bt
6	20	20	16	19	16	16	42	56	72	11	20		3	98 Bt
7	20	20	15	18	16	16	48	70	65	20	20		3	97 Bt
8	20	20	16	19	16	16	32	60	41	19	20		2	100 Bt
9	20	20	12	17	11	16	38	50	63	16	20		3	86 Bt
10	20	20	16	19	16	16	30	40	44	6	19		1	94 U
11	20	20	15	19	13	16	39	43	39	10	20		1	97 B
12	20	20	16	18	14	16	35	50	50	20	20		3	98 B
13	20	20	16	19	16	16	42	68	75	20	20		3	100 Bt
14	20	20	16	19	16	16	49	55	66	16	20		3	100 Bt
15	20	20	16	19	16	16	41	50	62	19	20		3	100 Bt
16	20	20	15	18	13	16	44	60	50	16	20		3	95 Bt
17	20	20	16	18	12	16	47	56	43	16	20		3	92 A
18	20	20	16	17	13	16	46	58	46	19	20		2	95 B
19	20	20	12	16	14	14	38	38	27	10	19		1	86 U
20	20	20	16	19	16	16	40	55	22	20	20		2	97 B
21	20	20	15	19	12	16	35	44	14	8	16		1	95 U
22	20	20	16	19	14	16	36	62	51	18	20		3	100 B
23	20	20	19	16	12	16	38	23	11	12	18		1	88 U
24	20	20	14	19	11	16	35	50	48	20	20		2	92 B
25	20	20	16	18	16	16	38	68	39	10	20		2	98 B

Name	Date
	Month
	Period

[illegible][illegible]

[illegible]

RECORD OF CLASS WORK						
Unit No.	Date	TOPICS COVERED	PERIOD	CUMULATIVE PERIOD	STAFF INITIAL	HOD INITIAL
II	5.7.19	stack ADT	7	10	Kade	S. P. 12/7/19
	8.7.19	Operations	1	11	Kade	
	9.7.19	stack - Applications	2	12	Kade	
	10.7.19	class test-1, stack ADT	5	13	Kade	
	11.7.19	Evaluating arithmetic expressions	2	14	Kade	
	12.7.19	Conversion of infix to post fix exp	7	15	Kade	
	13.7.19	Queue ADT	1	16	Kade	
	15.7.19	Operations	1	17	Kade	
	16.7.19	Circular Queue	5	18	Kade	
		Virtual Lab Session (stack Appln)				
	18.7.19	Priority Queue	2	19	Kade	
	19.7.19	Deque, Application of Queue	8	20	Kade	
Hours Planned: 10			Hours Handled: 11			

RECORD OF CLASS WORK						
Unit No.	Date	TOPICS COVERED	PERIOD	CUMULATIVE PERIOD	STAFF INITIAL	HOD INITIAL
III	23.7.19	Tree ADT	5	21	Kob	S/S
	23.7.19	Tree Traversal	6	22	Kob	
	26.7.19	Binary Tree ADT	6	23	Kob	
	27.7.19	Expression Trees	3	24	Kob	
		Application of Trees	4	25	Kob	
	29.7.19	Binary Search Tree	1	26	Kob	
	30.7.19	Threaded Binary Trees	5	27	Kob	
		AVL Trees	6	28	Kob	
	1.8.19	B-Trees	2	29	Kob	
	3.8.19	B+ Trees	1	30	Kob	
	5.8.19	Heap, Applications of Heap	1	31	Kob	
Hours Planned: 10			Hours Handled: 11			

[illegible]

RECORD OF CLASS WORK						
Unit No.	Date	TOPICS COVERED	PERIOD	CUMULATIVE PERIOD	STAFF INITIAL	HOD INITIAL
V	24.8.19	Sorting - Bubble Sort	2	42	Ked	
		Searching, Linear Search	7	43	Ked	
	26.8.19	Insertion Sort, Selection Sort	1	44	Ked	
	27.8.19	Radix Sort	5	45	Ked	
	30.8.19	Hashing, Hashing Functions	7	46	Ked	
	30.8.19	Separate chaining, open Addressing	8	47	Ked	
	31.8.19	Rehashing, Extensible Hashing	5	48	Ked	
	1.9.19	Extensible Hashing	5	49	Ked	
		open Addressing	6	50	Ked	
	14.9.19	Revision - Searching?	1	51	Ked	
	18.9.19	Radix Sort, Insertion Sort	1	52	Ked	
Hours Planned: 10			Hours Handled: 9+2(R)			

[illegible]



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SUBJECT: DATA STRUCTURES

SEMESTER: III

QUESTION BANK (CS8391)
(Version : 2)

PREPARED BY
Ms.K.ABHIRAMI / CSE

*Verified
f. 1/1/19.*

CS8391

DATA STRUCTURES

LT PC
3003

UNIT I - LINEAR DATA STRUCTURES - LIST

9

Abstract Data Types (ADTs) - List ADT - array-based implementation - linked list implementation - singly linked lists- circularly linked lists- doubly-linked lists - applications of lists -Polynomial Manipulation - All operations (Insertion, Deletion, Merge, Traversal).

UNIT II - LINEAR DATA STRUCTURES - STACKS, QUEUES

9

Stack ADT - Operations - Applications - Evaluating arithmetic expressions- Conversion of Infix to postfix expression - Queue ADT - Operations - Circular Queue - Priority Queue - deQueue - applications of queues.

UNIT III - NON LINEAR DATA STRUCTURES - TREES

9

Tree ADT - tree traversals - Binary Tree ADT - expression trees - applications of trees - binary search tree ADT -Threaded Binary Trees- AVL Trees - B-Tree - B+ Tree - Heap - Applications of heap.

UNIT IV -NON LINEAR DATA STRUCTURES - GRAPHS

9

Definition - Representation of Graph - Types of graph - Breadth-first traversal - Depth-first traversal - Topological Sort - Bi-connectivity - Cut vertex - Euler circuits - Applications of graphs.

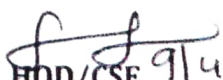
UNIT V- SEARCHING, SORTING AND HASHING TECHNIQUES

9

Searching- Linear Search - Binary Search. Sorting - Bubble sort - Selection sort - Insertion sort - Shell sort - Radix sort. Hashing- Hash Functions - Separate Chaining - Open Addressing - Rehashing - Extendible Hashing.

TOTAL: 45 PERIODS

 9/4/19
SIGNATURE OF STAFF INCHARGE

 9/4/19
HOD/CSE



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE PLAN

Sub. Code : CS8391	Branch / Year / Sem : B.E CSE / II / III
Sub.Name : Data Structures	Batch : 2018-2022
Staff Name : Ms.K..Abhirami	Academic Year : 2019-20 (ODD)

COURSE OBJECTIVE

1. To understand the concepts of ADTs
2. To Learn linear data structures – lists, stacks, and queues
3. To understand sorting, searching and hashing algorithms
4. To apply Tree and Graph structures

TEXT BOOKS

- T1.** Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", 2nd Edition, Pearson Education, 1997.
- T2.** Reema Thareja, "Data Structures Using C", Second Edition, Oxford University Press, 2011

REFERENCE BOOKS

- R1.** Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", Second Edition, McGraw Hill, 2002.
- R2.** Aho, Hopcroft and Ullman, "Data Structures and Algorithms", Pearson Education, 1983.
- R3.** Stephen G. Kochan, "Programming in C", 3rd edition, Pearson Education.
- R4.** Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, "Fundamentals of Data Structures in C", Second Edition, University Press, 2008

WEB RESOURCES

- W1.** <https://www.cse.iitkgp.ac.in/~pds/semester/2016a/JM/Wk11-linkedlist.ppt> (Topic.No:04)
- W2.** <https://www.cs.ust.hk/~huamin/COMP171/stack-queue.ppt> (Topic.No:16)
- W3.** <https://nptel.ac.in/courses/106102064/6> (Topic.No:22)
- W4.** www.csie.ntu.edu.tw/~ds/ppt/ch6/chapter6.PPT (Topic.No:34)
- W5.** https://www.cs.ccu.edu.tw/~tsaic/teaching/fall2005_gt/lecture09.ppt (Topic.No:40)
- W6.** <https://courses.cs.washington.edu/courses/cse326/RadixSort.ppt> (Topic.No:51)

Topic No	Topic	Books for Reference	Page No.	Teaching Methodology	No. of Hours Required	Cumulative No. of periods
UNIT I						
LINEAR DATA STRUCTURES – LIST						9+1
1.	Abstract Data Types (ADTs)	T1	57-58	BB	01	01
2.	List ADT	T1	58-58	BB		
3.	Array-based implementation	T1	59-59	BB	01	02
4.	Linked list implementation- Singly linked lists	T1, W1	59-61	PPT	02	04
5.	Circularly linked lists	T1	68-68	PPT	01	05
6.	Doubly-linked lists	T1	67-68	BB	01	06
7.	Applications of lists	T1	73-78	BB	01	07
8.	Polynomial Manipulation	T1	68-70	BB	01	08
9.	All operations (Insertion, Deletion, Merge, Traversal)	T1	61-65	BB	02	10
LEARNING OUTCOME						
At the end of unit, students will be able to						
• Gain knowledge on Abstract data type & Linked lists • Implement the operations on various lists.						
UNIT II						
LINEAR DATA STRUCTURES – STACKS, QUEUES						9+1
10.	Stack ADT	T1	78-79	BB	01	11
11.	Operations	T1	79-86	BB	01	12
12.	Applications	T1	87-87	BB	01	13
13.	Evaluating arithmetic expressions	T1	87-90	BB	01	14
14.	Conversion of Infix to postfix expression	T1	90-93	BB	01	15
15.	Queue ADT	T1	95-95	PPT	01	16
16.	Operations	T1, W2	96-99			
17.	Circular Queue	T2 R2	260-263 95-99	BB	01	17
18.	Priority Queue	T2	268-271	BB	01	18
19.	DeQueue	T2	264-267	BB	01	19
20.	Applications of queues	T1	100-101	Sem	01	20

Topic No	Topic	Books for Reference	Page No.	Teaching Methodology	No. of Hours Required	Cumulative No. of periods
LEARNING OUTCOME						
At the end of unit, students will be able to						
- Understand about Linear data structures Stacks and Queues. - Implement the operations on Stack and Queue. - Identify appropriate data structures to handle data.						
UNIT III		NON LINEAR DATA STRUCTURES – TREES				9+1
21.	Tree ADT	T1	106-106	BB	01	21
22.	Tree traversals	T1, W3	107-111	NPTEL	02	23
23.	Binary Tree ADT	T1	111-112			
24.	Expression trees	T1	113-116			
25.	Applications of trees	T2	294-296	BB	01	24
26.	Binary search tree ADT	T1	116-117	BB	01	25
27.	Threaded Binary Trees	T2	311-315	BB	01	26
28.	AVL Trees	T1	126-131	BB	01	27
29.	B-Tree	T1	149-152	BB	01	28
30.	B+ Tree	T1 R3	152 -154 150-153	BB	01	29
31.	Heap	T1	193--194	BB	01	30
32.	Applications of heap	T1	205-207			
LEARNING OUTCOME						
At the end of unit, students will be able to						
- Understand various non linear data structures - Implement various operations on trees. - Appraise the application of trees.						
UNIT IV		NON LINEAR DATA STRUCTURES - GRAPHS				9+1
33.	Definition	T1	200-300	BB	01	31
34.	Representation of Graph	T1, W4	300-302	PPT		
35.	Types of graph	T2	384-386	BB	01	32
36.	Breadth-first traversal	T2	393-396	BB	01	33
37.	Depth-first traversal	T2	397-399	BB	01	34
38.	Topological Sort	T1	302-306	BB	01	35
39.	Bi-connectivity	T1	338-342	BB	01	36
40.	Cut vertex	R1, W5	220-224	PPT	01	37

Topic No	Topic	Books for Reference	Page No.	Teaching Methodology	No. of Hours Required	Cumulative No. of periods
41.	Euler circuits	T1	342-345	BB	01	38
42.	Applications of graphs	T2	419-423	BB	02	40

LEARNING OUTCOME

At the end of unit, students will be able to

- Gain knowledge about non-linear data structures.
- Implement the operations on graphs.

UNIT V SEARCHING, SORTING AND HASHING TECHNIQUES 9+1

43.	Searching	T2	424-424	BB	01	41
44.	Linear Search	T2	424-426	BB		
45.	Binary Search	T2	426-428	BB	01	42
46.	Sorting	T1	235-236	BB	01	43
47.	Bubble sort	T2	434-437			
48.	Selection sort	T2	440-442	BB	01	44
49.	Insertion sort	T1	236-237	BB	01	45
50.	Shell sort	T1	238-240	CBT	01	46
51.	Radix sort	T2,W6	450-543			
52.	Hashing	T1	165-165	BB	01	47
53.	Hash Functions	T1	166-168			
54.	Separate Chaining	T1	168-172	BB	01	48
55.	Open Addressing	T1	173-180	BB		
56.	Rehashing	T1 R4	181-183 331-334	BB	01	49
57.	Extendible Hashing	T1	184-190	BB	01	50

LEARNING OUTCOME

At the end of unit, students should be able to

- Gain knowledge about sorting and searching algorithms.
- Implement sorting and hashing techniques.

COURSE OUTCOME

At the end of the course, the students will be able to

1. Implement abstract data types for linear data structures.
2. Apply the different linear and non-linear data structures to problem solutions.
3. Critically analyze the various sorting algorithms.

CONTENT BEYOND THE SYLLABUS

1. Spanning trees (Virtual Lab Session)

INTERNAL ASSESSMENT DETAILS

ASST. NO.	I	II	MODEL
Topic Nos.	1-14	15-32	1-57
Date			

ASSIGNMENT DETAILS

ASSIGNMENT	I	II
Topic Nos. for reference / Activity	1-14	PCE
Deadline	Before ATI	Before ATII

ASSIGNMENT I (20 marks) BEFORE AT-I	ASSIGNMENT II (20 marks) BEFORE AT-II																																	
(Written Assignment)	PCE Skills Activity																																	
Part A 1. List down any four applications of data structures. 2. Define linked lists. 3. Give few examples for data structures 4. What are the three ways to represent arithmetic expression? 5. Explain the usage of stack in recursive algorithm implementation. Part B 1. a. Explain in detail three applications of linked list with suitable example b. Write routines to implement addition, subtraction & differentiation of two polynomials. 2. Explain the array and linked list implementation of Stack. Illustrate the application of stack. 3. Write a procedure to convert the given infix expression to postfix expression and postfix to infix expression using stack.	Activity 1: Application of concept -Consider five cities: (1) New Delhi, (2) Mumbai, (3) Chennai, (4) Bangalore, and (5) Kolkata, and a list of flights that connect these cities as shown in the following table. Use the given information to construct a graph and explain.(10) <table><tr><th>Flight No.</th><th>Origin</th><th>Destination</th></tr><tr><td>101</td><td>2</td><td>3</td></tr><tr><td>102</td><td>3</td><td>2</td></tr><tr><td>103</td><td>5</td><td>3</td></tr><tr><td>104</td><td>3</td><td>4</td></tr><tr><td>105</td><td>2</td><td>5</td></tr><tr><td>106</td><td>5</td><td>2</td></tr><tr><td>107</td><td>5</td><td>1</td></tr><tr><td>108</td><td>1</td><td>4</td></tr><tr><td>109</td><td>5</td><td>4</td></tr><tr><td>110</td><td>4</td><td>5</td></tr></table> Activity 2: GATE questionnaire Activity 3: Concept Map – Sorting Algorithm Activity 4: Virtual Lab exercise – Expression trees Activity 5: Coding exercise – Trees & Graphs Activity 6: Google classroom - Quiz	Flight No.	Origin	Destination	101	2	3	102	3	2	103	5	3	104	3	4	105	2	5	106	5	2	107	5	1	108	1	4	109	5	4	110	4	5
Flight No.	Origin	Destination																																
101	2	3																																
102	3	2																																
103	5	3																																
104	3	4																																
105	2	5																																
106	5	2																																
107	5	1																																
108	1	4																																
109	5	4																																
110	4	5																																

K. Abhirami
 Prepared by 9/4/19
 Ms.K.Abhirami

J. Ramesh
 23/4/19
 Approved by
 PRINCIPAL

J. Ramesh
 Verified By 23/4/19
 HOD/CSE



KINGS
COLLEGE OF ENGINEERING



TÜVRheinland
CERT
ISO 9001

PUNALKULAM, THANJAVUR - 613 303

NAAC ACCREDITED INSTITUTION

Approved by AICTE, New Delhi and Affiliated to Anna University, Chennai

ACADEMIC YEAR 2019 - 2020 / ODD SEMESTER

QUESTION BANK

Name of the Student : _____

Register Number : _____

Year / Semester : _____



KINGS
COLLEGE OF ENGINEERING
(NAAC Accredited Institution)
(Approved by AICTE, New Delhi, Affiliated to
Anna University, Chennai)



TÜV Rheinland
CERT
ISO 9001

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ACADEMIC YEAR 2019 – 2020/ ODD SEMESTER

TABLE OF CONTENTS

S. NO.	SUBJECT CODE	NAME OF THE SUBJECT	CONTENTS	PAGE NO.
1	MA8351	Discrete Mathematics	Syllabus Course Plan Question Bank University Question Papers	DM 2 DM 3 DM 8 DM 35
2	CS8351	Digital Principles and System Design	Syllabus Course Plan Question Bank	DPSD 2.2 DPSD 2.3 DPSD 2.8
3	CS8391	Data structures	Syllabus Course Plan Question Bank University Question Papers	DS 3.2 DS 3.3 DS 3.9 DS 3.47
4	CS8392	Object Oriented Programming	Syllabus Course Plan Question Bank	OOP 4.2 OOP 4.3 OOP 4.9
5	EC8395	Communication Engineering	Syllabus Course Plan Question Bank	CE 5.2 CE 5.3 CE 5.7



KINGS
COLLEGE OF ENGINEERING
(NAAC Accredited Institution)
(Approved by AICTE, New Delhi, Affiliated to
Anna University, Chennai)



TOURNAMENT
CERT
ISO 9001

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SUBJECT: DATA STRUCTURES

SEMESTER: III

QUESTION BANK (CS8391)
(Version : 2)

PREPARED BY
Ms.K.ABHIRAMI / CSE



UNIT I - LINEAR DATA STRUCTURES

PART- A

1. State the advantage of ADT

(AU-ND 2018)

- ADT is reusable and ensures robust data structure.
- It reduces coding efforts.
- Encapsulation ensures that data cannot be corrupted.
- ADT is based on principles of Object Oriented Programming (OOP) and Software Engineering (SE).
- It specifies error conditions associated with operations.

2. What are the disadvantage of linked list over array?

(AU-ND 2018)

- Random access is not allowed. We have to access elements sequentially starting from the first node. So we cannot do a binary search with linked lists.
- Extra memory space for a pointer is required with each element of the list.
- Arrays have better cache locality that can make a pretty big difference in performance.

3. What is called as Data Structure.

- Data structure is an **arrangement of data in computer's memory**. It makes the data quickly available to the processor for required operations.
- It is a software artifact which **allows data to be stored, organized and accessed**.
- It is a structure program used to store ordered data, so that various operations can be performed on it easily.

4. Differentiate between data type and data structure.

Data Type: Data Type of a variable is a **set of values the variable may assume**. Eg. int, char & real

Data Structures: Data structure mean the **way of organizing data values** with the help of existing data types. Eg. Array, stack queue, & tree.

5. State the significance of data structure.

- Data structure is important because it is **used in almost every program or software system**.
- It helps **to write efficient code, structures the code and solve problems**.
- **Data can be maintained more easily** by encouraging a better design or implementation.

- Data structure is just a container for the data that is **used to store, manipulate and arrange**. It can be processed by algorithms.

PART-B

- 1.i. State the polynomial representation for $6x^3+9x^2+7x+1$ using linked list.
Write procedure to add and multiply two polynomial and explain with suitable example. (7)
AU-ND 2018(7)
- ii. What are the ways to insert a node in linked list?. Write an algorithm for inserting a node before a given node in a linked list. (6)
AU-ND 2018(6)
- 2.i. What are the various operations on array?. Write a procedure to insert an element in the middle of the array. (7)
AU-ND 2018(7)
- ii. Write a procedure to delete the last node from a circular linked list. (6)
AU-ND 2018 (6)
- 3.i. Develop an algorithm to merge two sorted linked list into a single sorted list. (7)
- ii. Explain the operation of inserting an element at the front, middle and rear in a doubly linked list. (6)
4. Describe the routines to perform the following operations on linked list: Create, Insert, Delete, Traverse, Search, Concatenation and display. (13)
- 5.i. Explain about polynomial manipulation with example (7)
- ii. Enumerate on the application of lists. (6)
6. Develop an algorithm to perform the following operations in a doubly linked list
Insert a node at the end of the list.
Delete the last node in the list.
Search an element in the list. (13)
7. Develop an algorithm for deletion and traversal operation on doubly linked list (13)
8. Construct an algorithm for creation and insertion operation on circular linked list. (13)
9. Consider an array A[1: n] Given a position, write an algorithm to insert an element in the Array. If the position is empty, the element is inserted easily. If the position is already occupied the element should be inserted with the minimum number of shifts. (Note: The elements can shift to the left or to the right to make the minimum number of moves). (13)
10. Develop an algorithm to add the values of the nodes in a linked list and calculate the mean (13)
11. Explain the steps involved in the following insertion operations on a singly linked list.
Insert the node in the start and end. (7)
Insert the node in the middle of the list. (6)
12. Explain an algorithm to split a linked list into two sub lists containing odd and even ordered element in them respectively. (13)
13. Describe the various operations of the list ADT with examples (13)
14. Illustrate how polynomial manipulations are performed using lists?. Explain with suitable example. (13)

15. Discuss the procedure for cursor based linked list implementation and to perform various operations on it. (13)

UNIT II – LINEAR DATA STRUCTURES – STACKS, QUEUES

PART – A

1. What are the application of stacks?. (AU-ND 2018)

Following are the applications of stack:

- Expression Evaluation
- Expression Conversion
 - Infix to Postfix
 - Infix to Prefix
 - Postfix to Infix
 - Prefix to Infix
- Backtracking
- Memory Management

2. What are priority queues?. What are the ways to implement priority queue?. (AU-ND 2018)

Priority Queue is an extension of queue with following properties.

- Every item has a priority associated with it.
- An element with high priority is dequeued before an element with low priority.
- If two elements have the same priority, they are served according to their order in the queue.
- Priority queue can be implement
 - Using array
 - Using heaps

3. Explain stack data structure in brief.

- Stack is an ordered list of the same type of elements.
- It is a linear list where all insertions and deletions are permitted only at one end of the list.
- Stack is a LIFO (Last In First Out) structure.
- In a stack, when an element is added, it goes to the top of the stack.
- When a stack is completely full, it is said to be **Overflow state** and if stack is completely empty, it is said to be **Underflow state**.

4. List the operations that can be performed in a stack. Explain briefly. / State stack ADT

A Stack contains elements of same type arranged in sequential order. All operations takes place at a single end that is top of the stack and following operations can be performed:

There are two basic operations performed in a Stack:

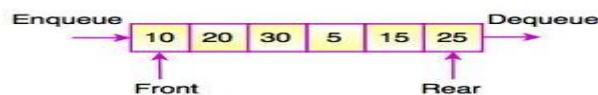
- Push() - is used to add or insert new elements into the stack.
- Pop() - is used to delete or remove an element from the stack.

Other operations that can be performed in a stack are:

- peek() – gets the top element of the stack without deleting it.
- isempty() – checks whether the stack is empty or not
- isnull() – used to check whether stack is full or not.
- Size() – returns the number of elements in the stack.

5. What is Queue?.

- Queue is a linear data structure where the first element is inserted from one end called **REAR** and deleted from the other end called as **FRONT**.
- **Front** points to the **beginning** of the queue and **Rear** points to the **end** of the queue.
- Queue follows the **FIFO (First - In - First Out)** structure.
- According to its FIFO structure, element inserted first will also be removed first.
- In a queue, one end is always used to insert data (enqueue) and the other is used to delete data (dequeue), because queue is open at both its ends.



6. List the types of Queue.

There are four types of Queue.

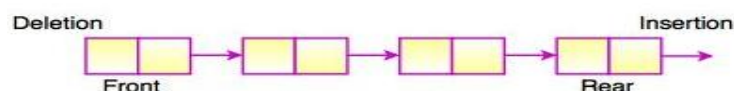
- Simple queue
- Circular queue
- Priority queue
- Dequeue (double ended queue)

7. Tabulate the operations that can be performed on queue.

Operations	Description
enqueue()	This function defines the operation for adding an element into queue
dequeue()	This function defines the operation for removing an element from queue
init()	This function is used for initializing the queue.
front	Front is used to get the front data item from a queue
rear	Rear is used to get the last item from a queue.

8. Write short note on Simple Queue.

Simple queue defines the simple operation of queue in which insertion occurs at the rear of the list and deletion occurs at the front of the list.



PART-B

1. Write the procedure to convert the infix expression to postfix expression and steps involved in evaluating the postfix expression. Convert the expression $A - (B/C + (D\%E * F)/G) * H$ to postfix form. Evaluate the given postfix expression $9\ 3\ 4\ * \ 8\ + \ 4\ / \ -$ (AU-ND 2018)
2. What are circular queues. Write the procedure to insert an element to circular queue and delete an element from a circular queue using array implementation. (AU-ND 2018)
- 3.a. Describe with an example how to evaluate arithmetic expression using stacks (7)
b. Explain about stack ADT in detail. (6)
- 4.a. Describe the procedure for array based implementation of stack (7)
b. Explain linked list based implementation of stack. (6)
- 5.a. Write the algorithm to convert an infix expression to a postfix expression. Trace the algorithm to convert infix expression $(a+b)*c/d+e/f$ to a postfix expression. (8)
b. Justify the need for infix and postfix expression. (5)
- 6.a. Give an algorithm for push and pop operation on stack using linked list. (7)
b. Discuss about addition and deletion operation on a circular queue with necessary algorithms. (6)
- 7.a. Illustrate the evaluation of postfix expression. (7)
b. Describe process of converting infix expression to postfix expression using stack (6)
- 8.a. Write an algorithm that checks if expression is correctly parenthesized using stack and illustrate with an example. (7)
b. Write the function to examine whether the stack is full() or empty(). (6)
- 9.a. Describe about Queue ADT in detail. (7)
b. Explain any application of queue with suitable example. (6)
10. Explain about the operations of Queue with example. (13)
11. Analyze and write algorithm to implement queue function using array. (13)
12. Develop an algorithm to perform the operations in a doubly ended queue implemented as an array. (13)
13. Illustrate Circular queue implementation and its operations. (13)
14. Discuss priority queue implementation, its operation and application. (13)
15. Discuss the procedure and trace the evaluation of the expression using stack $12+3*4-(5*16)+7(15)$ (13)

UNIT III – NON LINEAR DATA STRUCTURES – TREES

PART – A

1. Show the result of in-order traversal of the binary search tree given in figure 2.

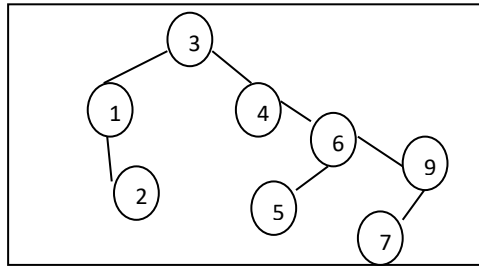


Fig.1

In-order traversal of the tree : 1 2 3 4 5 6 7 9

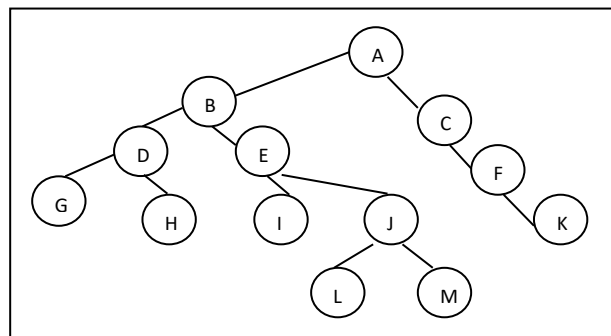
(AU-ND 2018)

2. For the tree in Figure 2.

- List the siblings for node E
- Compute the height

(AU-ND 2018)

FIG.2

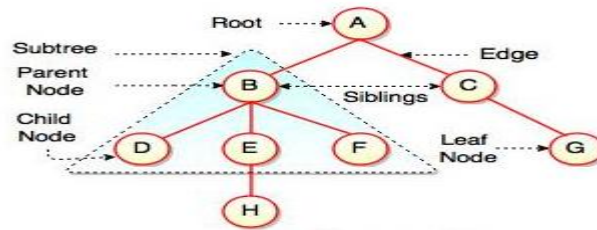


Siblings for node E is D, Height of the given tree is 5

3. Write short note on Tree data structure.

- Tree is a hierarchical data structure which stores the information naturally in the form of hierarchy style.
- Tree is one of the most powerful and advanced data structures.
- It is a non-linear data structure compared to arrays, linked lists, stack and queue.
- It represents the nodes connected by edges.

4. Schematically represent tree structure with its various entities involved.



5. Explain height of node & tree, depth of node, degree of node and edge of a tree.

Height of Node	Height of a node represents the number of edges on the longest path between that node and a leaf.
Height of Tree	Height of tree represents the height of its root node.
Depth of Node	Depth of a node represents the number of edges from the tree's root node to the node.
Degree of Node	Degree of a node represents a number of children of a node.
Edge	Edge is a connection between one node to another. It is a line between two nodes or a node and a leaf.

6. List the advantage of Tree data structure.

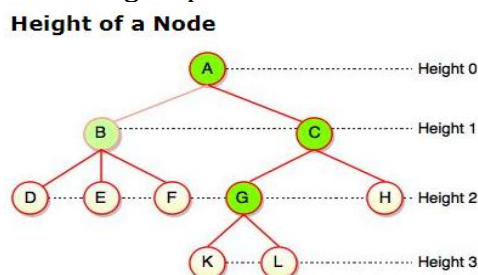
- Tree reflects structural relationships in the data.
- It is used to represent hierarchies.
- It provides an efficient insertion and searching operations.
- Trees are flexible. It allows to move subtrees around with minimum effort.

7. Explain External and Internal node of a tree.

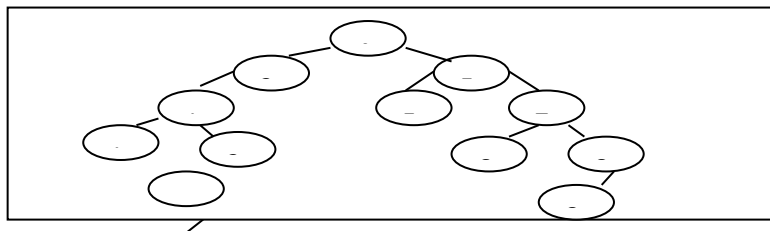
- If node has no children, it is called **Leaves** or **External Nodes**.
- Nodes which are not leaves, are called **Internal Nodes**. Internal nodes have at least one child.
- A tree can be empty with no nodes or a tree consists of one node called the **Root**.

8. Schematically represent height of a tree.

Height of a node defines the longest path from the node to a leaf.



1. Write the following routines to implement the basic binary search tree operations.
 - a. Perform search operation in binary search tree.
 - b. Find_min and Find_max (AU-ND 2018)
2. Distinguish between B Tree and B+ tree. Create a B tree of order 5 by inserting the following elements: 3,14,7,1,8,5,11,17,13,6,23,12,20,26,4,16,24,25,19. (AU-ND 2018)
3. Write an algorithm to perform preorder, inorder and postorder traversal on a binary tree. (13)
4. Explain insertion and deletion operation on a binary search tree with suitable algorithms. (13)
- 5.a. Write short notes on threaded binary tree. (7)
 - b. Describe an iterative algorithm to traverse a tree in preorder. (6)
6. Discuss in detail the various methods in which a binary tree can be represented. Discuss the advantage and disadvantage of each method. (13)
- 7.a. Explain B+ tree and its properties with example. (7)
 - b. What are the steps to convert general tree to binary tree. (6)
- 8.a. Construct B tree to insert the following key elements (order of the tree is 3) 5,2,13,3,45,72,4,6,9,22. (7)
 - b. Draw a B tree of order 6. (6)
9. a. What are AVL trees?. Describe the different rotations defined for AVL tree. (7)
 - b. Insert the following elements in sequence into an empty AL tree step by step. 15, 18, 20, 21, 28, 23, 30, 26 (6)
- 10.a. Point out the operations of B-tree using 2-3 tree. (7)
 - b. Explain the operations of threaded binary tree. (6)
- 11.a. Show the result of inserting 15, 17, 6, 19, 11, 10, 13, 20, 8, 14, 12 one at a time into an initially empty binary min heap. (7)
 - b. Show the result of performing minimum three delete operations in binary min heap obtained. (6)
- 12.a. Illustrate how delete operation is performed in binary heap. (7)
 - b. Write suitable procedure for procolate up and percolate down operations in a binary heap. (6)
- 13.a. Explain the construction of expression tree with example. (7)
 - b. Give the application of tree. (6)
- 14.a. Compare B tree with B+ tree. (7)
 - b. Create a B+ tree of order 5 for the following data arriving in sequence. 90, 27, 7, 9, 18, 21, 3, 4, 16, 11, 21, 72. (8)
- 15.a. Consider the binary search tree given below. Find the result of in-order, pre-order and post-order traversal. Show the deletion of the root node. Insert 11, 22, 33, 44, 55 and 77 in the tree.



UNIT IV – NON LINEAR DATA STRUCTURES – GRAPHS

PART – A

1. What are the representation of the graph? (AU-ND 2018)

Following two are the most commonly used representations of a graph.

- Adjacency Matrix
- Adjacency List

There are other representations also like, Incidence Matrix and Incidence List. The choice of the graph representation is situation specific. It totally depends on the type of operations to be performed and ease of use.

2. Define Euler circuits (AU-ND 2018)

Eulerian Path is a path in graph that visits every edge exactly once. Eulerian Circuit is an Eulerian Path which starts and ends on the same vertex. A graph is called Eulerian if it has an Eulerian Cycle and called Semi-Eulerian if it has an Eulerian Path.

3. Write short note on Graph data structure.

Graph is a data structure that consists of following two components:

- A finite set of vertices also called as nodes.
- A finite set of ordered pair of the form (u, v) called as edge.

The pair is ordered because (u, v) is not same as (v, u) in case of a directed graph (di-graph). The pair of the form (u, v) indicates that there is an edge from vertex u to vertex v . The edges may contain weight/value/cost

4. List the application of graph.

Graphs are used to represent many real-life applications:

- Graphs are used to represent networks. The networks may include paths in a city or telephone network or circuit network.
- Graphs are also used in social networks like linkedIn, Facebook. For example, in Facebook, each person is represented with a vertex (or node). Each node is a structure and contains information like person id, name, gender and locale.

5. What is a Directed Graph?.

- If a graph contains ordered pair of vertices, is said to be a Directed Graph.
- If an edge is represented using a pair of vertices (V_1, V_2) , the edge is said to be directed from V_1 to V_2 .
- The first element of the pair V_1 is called the start vertex and the second element of the pair V_2 is called the end vertex.

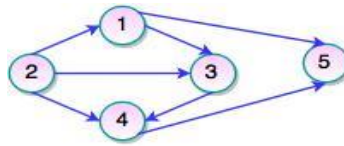


Fig. Directed Graph

Set of Vertices $V = \{1, 2, 3, 4, 5\}$

Set of Edges $W = \{(1, 3), (1, 5), (2, 1), (2, 3), (2, 4), (3, 4), (4, 5)\}$

6. Explain in brief about Undirected graph.

- If a graph contains unordered pair of vertices, is said to be an Undirected Graph.
- In this graph, pair of vertices represents the same edge.

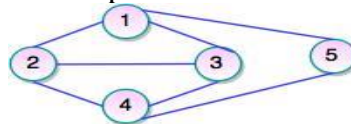
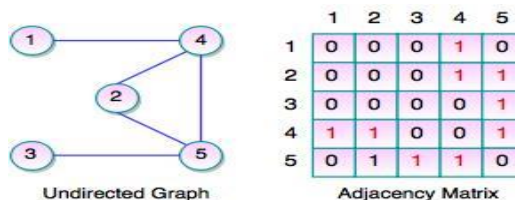


Fig. Undirected Graph

7. Explain about Adjacency matrix representation of graph.

- Adjacency matrix is a way to represent a graph.
- It shows which nodes are adjacent to one another.
- Graph is represented using a square matrix.



Undirected Graph

	1	2	3	4	5
1	0	0	0	1	0
2	0	0	0	1	1
3	0	0	0	0	1
4	1	1	0	0	1
5	0	1	1	1	0

Adjacency Matrix

Fig. Adjacency Matrix Representation of Undirected Graph

8. State the advantages of adjacency matrix way of graph representation.

Advantages of Adjacency Matrix

- Adjacency matrix representation of graph is very simple to implement.
- Adding or removing time of an edge can be done in $O(1)$ time. Same time is required to check, if there is an edge between two vertices.
- It is very convenient and simple to program.

9. Mention the limitations of adjacency matrix

Limitations of Adjacency Matrix

- It consumes huge amount of memory for storing big graphs.
- It requires huge efforts for adding or removing a vertex. If you are constructing a graph in dynamic structure, adjacency matrix is quite slow for big graphs

10. What is a Adjacency list graph representation?.

- Adjacency list is a representation of graphs.
- It is a collection of unordered list, used to represent a finite graphs.
- Each list describes the set of neighbors of a vertex in the graph.

- Adjacency list requires less amount of memory.
- For every vertex, adjacency list stores a list of vertices, which are adjacent to the current one.
- In adjacency list, an array of linked list is used. Size of the array is equal to the number of vertices.

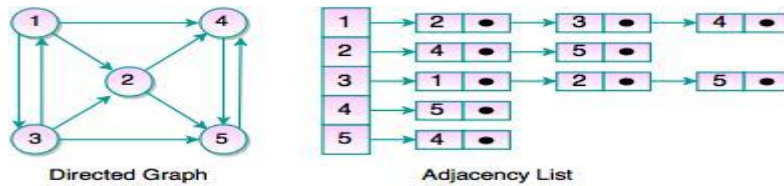
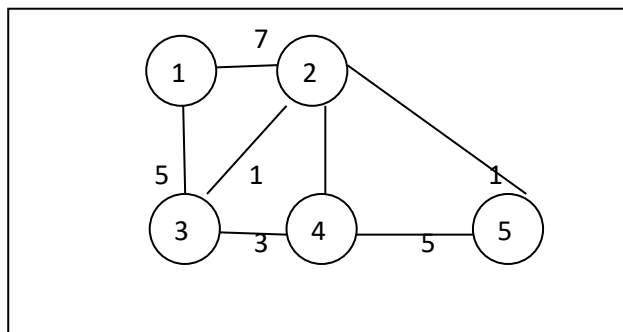


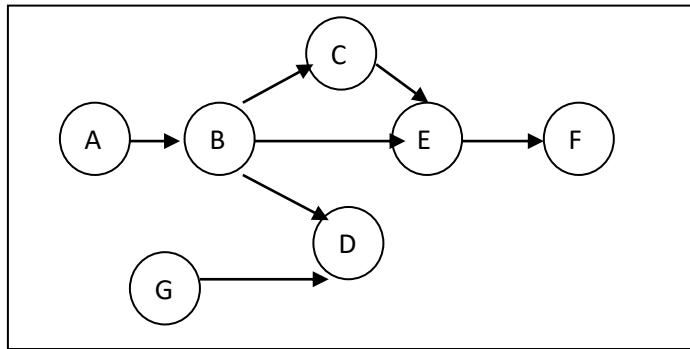
Fig. Adjacency List Representation of Directed Graph

PART-B

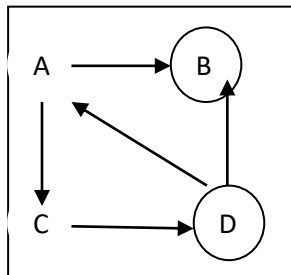
1. Distinguish between breadth first search and depth first search with example. (AU-ND 2018)
2. State and explain topological sort with suitable example. (AU-ND 2018)
3. Describe in detail about the following representations of a graph.
 - a. Adjacency matrix (7)
 - b. Adjacency list. (6)
4. a. Explain the topological sorting of a graph G with example. (7)
 - b. Elaborate on the stepwise procedure for topological sort. (6)
5. a. Explain with algorithm, how DFS be performed on a undirected graph. (7)
 - b. Show the algorithm for finding connected components of an undirected graph using DFS, and derive the time complexity of the algorithm. (6)
6. a. Discuss the algorithm for Breadth First Search on a graph. (7)
 - b. Illustrate with suitable example Depth First search on a graph. (6)
7. a. Illustrate Kruskal's algorithm to find the minimum spanning tree of a graph. (7)
 - b. Trace the algorithm for the following graph. (6)



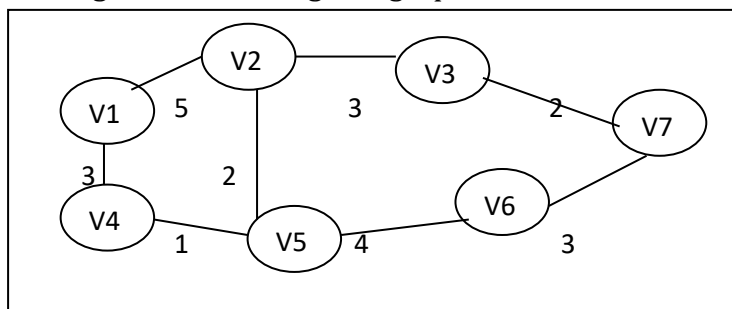
8. a. Consider the given directed acyclic graph D. Sort the nodes D by applying topological sort on D. (7)



b. Consider the graph given below and show its adjacency list in the memory



9. Compare any two application of graph with your own example. (13)
10. Discuss any one of the shortest path algorithm with suitable example. (13)
11. Discuss the prims algorithm for minimum spanning tree, Give an example. (13)
12. a. Write the procedure to find an Euler Circuit in a graph. (7)
- b. Trace the algorithm for the given graph. (6)



13. Explain the procedure to compute the shortest path using Dijkstra's algorithm. Validate the algorithm with suitable example. (13)
14. Explain the depth first approach of finding articulation points in a connected graph with necessary algorithm. (13)
15. a. Write short notes on Bi-connectivity. (7)
- b. Illustrate the different types of graphs with example. (6)
16. Given the adjacency matrix of a graph, write a program to calculate the in-degree and out-degree of a node N in the graph. (13)
17. How to check whether or not a graph is disconnected?. Describe an algorithm that uses brute force approach to find all the articulation points in G in $O(C(V+E))$ time. (13)
18. Given a rooted tree, one desires to find the shortest path from the root to a given node v. Which algorithm would one use to find the shortest path?. Write a program

to determine whether there is at least one path from the source to the destination. (13)

19. Consider 5 cities: New Delhi(1), Mumbai(2), Chennai(3), Bangalore(4) and Kolkata(5) and a list of flights that connect these cities as shown in the following table. Use the information to construct a graph. (13)

Flight No.	Origin	Destination
101	2	3
102	3	2
103	5	3
104	3	4
105	2	5
106	5	2
107	5	1
108	1	4
109	5	4
110	4	5

UNIT V – SEARCHING, SORTING AND HASHING TECHNIQUES

PART – A

1. What are the advantage and disadvantage of separate chaining and linear probing?.

Separate chaining

Pros

- Simple to implement
- Hash table capacity not limited
- Robust technique.
- Accommodate many keys

Cons

- Cache performance is not good
- Wastage of space
- Search time can be $O(n)$ in worst case
- Uses extra space for links

Linear probing

Pros

- Best cache performance
- Easy to compute
- Suffers from clustering

2. State the complexity of binary search.

Search a sorted array by repeatedly dividing the search interval in half. Begin with an interval covering the whole array. If the value of the search key is less than the item in the middle of the interval, narrow the interval to the lower half. Otherwise narrow it to the upper half. Repeatedly check until the value is found or the interval is empty.

The idea of binary search is to use the information that the array is sorted and reduce the time complexity to $O(\log n)$.

3. What is meant by collision?.

A hash function gets a small number for a key which is a big integer or string, there is a possibility that two keys result in the same value. The situation where a newly inserted key maps to an already occupied slot in the hash table is called collision. Collisions are very likely even if we have big table to store keys.

4. What is hashing?.

- Hashing is the process of mapping large amount of data item to smaller table with the help of hashing function.
- Hashing is also known as **Hashing Algorithm** or **Message Digest Function**.
- It is a technique to convert a range of key values into a range of indexes of an array.

5. How to handle collision?.

There are mainly two methods to handle collision:

- Separate Chaining - The idea is to make each cell of hash table point to a linked list of records that have same hash function value.
- Open Addressing - In Open Addressing, all elements are stored in the hash table itself. So at any point, size of the table must be greater than or equal to the total number of keys

6. List the advantages of Hashing.

- It is used to facilitate the next level searching method when compared with the linear or binary search.
- Hashing allows to update and retrieve any data entry in a constant time $O(1)$.
- Constant time $O(1)$ means the operation does not depend on the size of the data.
- Hashing is used with a database to enable items to be retrieved more quickly.
- It is used in the encryption and decryption of digital signatures.

7. Write short note on Hashing function.

- A fixed process converts a key to a hash key is known as a **Hash Function**.
- This function takes a key and maps it to a value of a certain length which is called a **Hash value** or **Hash**.

- Hash value represents the original string of characters, but it is normally smaller than the original.
- It transfers the digital signature and then both hash value and signature are sent to the receiver. Receiver uses the same hash function to generate the hash value and then compares it to that received with the message.
- If the hash values are same, the message is transmitted without errors.

PART-B

- 1.a. State and explain the shell sort. State and explain the algorithm for shell sort. Sort the elements using shell sort. (AU-ND 2018)(7)
- b. Explain Open Addressing in detail. (AU-ND 2018)(6)
- 2.a Distinguish between linear search and binary search. State and explain the algorithms for both the search with example. (AU-ND 2018)(13)
3. Explain Rehashing and extendible hashing.
Given input {4371, 1323, 6173, 4199, 4344, 9679, 1989} and a hash function $h(x) = x(\text{mod } 10)$, showing the resulting
 - i. open hash table
 - ii. closed hash table using linear probing
 - iii. closed hash table using quadratic probing
 - iv. closed.
4. Describe how the divide and conquer technique is implemented in binary search. (13)
5. Describe the algorithm to sort the following array: 77, 33, 44, 11, 88, 22, 66, 55
 - a.Insertion sort. (7)
 - b.Shell sort. (6)
6. List and illustrate the different types of hashing technique. (13)
- 7.a.Interpret the result of inserting the keys 2,3,5,7,11,13,15,6,4 into an initially empty extendible hashing data structure with $M=3$. (7)
- b.Discuss the running time of Divide-and-conquer merge sort algorithm. (6)
- 8.a.Sort the sequence 3,1,4,1,5,9,2,6,5 using Insertion sort. (7)
- b.Describe the routine for insertion sort. (6)
9. Write an algorithm to sort a set of N numbers using quick sort. Demonstrate the algorithm for the following set of number : 88,11,22,44,66,99,32,67,54,10 (13)
- 10.Explain the various collision resolution technique in detail with an example. (13)
- 11.Compare the Bucket sort and selection sort and discuss the methods. (13)
- 12.Sort the given integers and explain the intermediate results using shell sort.
35, 12, 14, 9, 15, 45, 32, 95, 40, 5. (13)
- 13.Illustrate Bubble sort technique and explain the algorithm. (13)
- 14.Explain Merge sort algorithm with example. (13)
- 15.Discuss the algorithm to search a number using binary search technique. (13)
- 16.Illustrate Extendible hashing technique. Explain in detail. (13)
- 17.Explain Selection sort and Radix sort algorithm with example. (13)

Question Paper Code : 250060
B.E / B.Tech Degree Examination, November/December 2018
Third Semester
Computer Science and Engineering
CS8391 – DATA STRUCTURES
(Regulation 2017)

Time : Three Hours

Maximum : 100 marks

Answer ALL Questions
Part – A (10*2=20 marks)

1. State the advantage of ADT.
2. What are the disadvantage of linked list over array?
3. What are the application of stacks?
4. What are priority queues?. What are the ways to implement priority queue?
5. For the tree in Figure 1.
 - List the siblings for node E
 - Compute the height

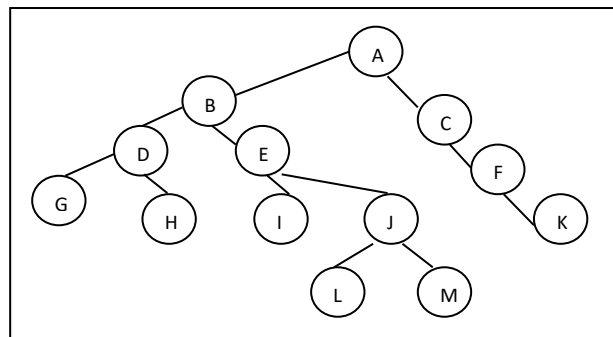


Fig.1

6. Show the result of in-order traversal of the binary search tree given in figure 2.

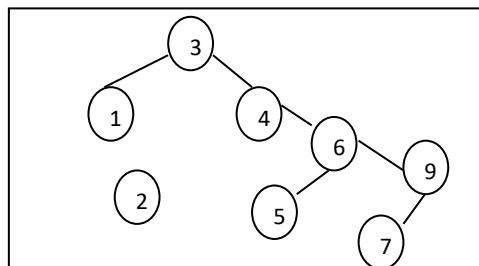


Fig.2

7. What are the representation of the graph?
8. Define Euler circuits
9. What are the advantage and disadvantage of separate chaining and linear probing?.
10. State the complexity of binary search.

Part – B (5*13=65 marks)

- 11.a.i. State the polynomial representation for $6x^3+9x^2+7x+1$ using linked list. Write procedure to add and multiply two polynomial and explain with suitable example. (7)
- ii. What are the ways to insert a node in linked list?. Write an algorithm for inserting a node before a given node in a linked list. (6)

Or

- b.i. What are the various operations on array?. Write a procedure to insert an element in the middle of the array. (7)
- ii. Write a procedure to delete the last node from a circular linked list. (6)

- 12.a. Write the procedure to convert the infix expression to postfix expression and steps involved in evaluating the postfix expression. Convert the expression $A-(B/C+(D\%E\%F)/G)*H$ to postfix form. Evaluate the given postfix expression $9\ 3\ 4\ *\ 8\ +\ 4\ /\ -$

Or

- 12.b. What are circular queues. Write the procedure to insert an element to circular queue and delete an element from a circular queue using array implementation.
- 13.a. Write the following routines to implement the basic binary search tree operations.
- i. Perform search operation in binary search tree.
- ii. Find_min and Find_max

or

- 13.b. Distinguish between B Tree and B+ tree. Create a B tree of order 5 by inserting the following elements: 3,14,7,1,8,5,11,17,13,6,23,12,20,26,4,16,24,25,19.
- 14.a. Distinguish between breadth first search and depth first search with example.

Or

- 14.b. State and explain topological sort with suitable example.
- 15.a.i. State and explain the shell sort. State and explain the algorithm for shell sort. Sort the elements using shell sort. (7)
- ii. Explain Open Addressing in detail. (6)

or

- 15.b.i. Distinguish between linear search and binary search. State and explain the algorithms for both the search with example. (7)
- ii. Explain Rehashing and extendible hashing. (6)

Part – C (1*15=15 marks)

16.a. What are expression trees. Writ the procedure for constructing an expression tree.

Or

16.b. Given input {4371, 1323, 6173, 4199, 4344, 9679, 1989} and a hash function $h(x) = x \pmod{10}$, showing the resulting

- open hash table
- closed hash table using linear probing
- closed hash table using quadratic probing
- closed.

UNIT - 1 - LINEAR DATA STRUCTURES - LIST

Data Structures – Introduction

A data structure is basically a group of data elements that are put together under one name, and which ***defines a particular way of storing and organizing data in a computer so that it can be used efficiently.***

Data structures are widely **applied** in the following areas:

- Compiler design
- Operating system
- Statistical analysis package
- DBMS
- Numerical analysis
- Simulation
- Artificial intelligence
- Graphics

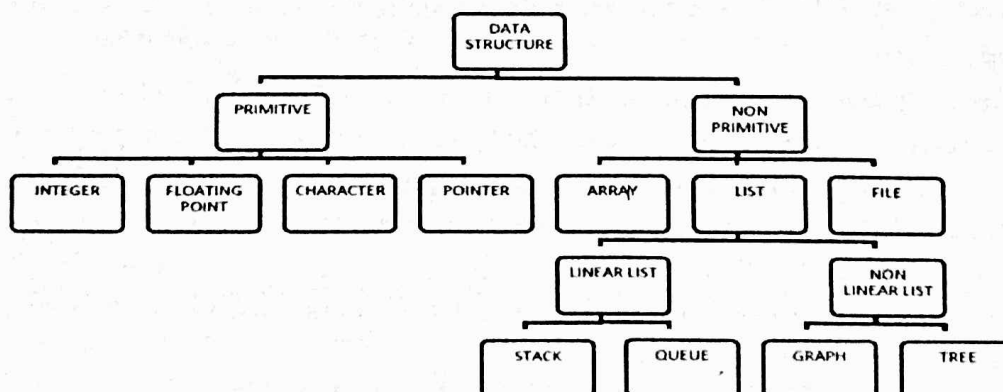
When selecting a data structure to solve a problem, the following steps must be performed.

- Analysis of the problem to determine the basic operations that must be supported. For example, basic operation may include inserting/deleting/searching a data item from the data structure.
- Quantify the resource constraints for each operation.
- Select the data structure that best meets these requirements

Significance of Data Structure

- Data structure is important because it is **used in almost every program or software system.**
- It helps **to write efficient code, structures the code and solve problems.**
- **Data can be maintained more easily** by encouraging a better design or implementation.
- Data structure is just a container for the data that is **used to store, manipulate and arrange.** It can be processed by algorithms.

Classification of Data Structure



Primitive

- Primitive data types are the **data types available in most of the programming languages.**
- These data types are used to **represent single value.**
- It is a **basic data type** available in most of the programming language.
- Eg : integer, float, character, Boolean.

Non-Primitive

- Data type **derived from primary data types** are known as Non-Primitive data types.
- Non-Primitive data types are used to **store group of values**.
- It is divided into Linear and Non-Linear data structure.

Linear & Non-Linear Data Structure

Feature	Linear Data structure	Non-Linear Data structure
Basic	Data items are arranged in an orderly manner where the elements are attached adjacently (sequentially)	It arranges the data in a sorted order and there exists a relationship between the data elements
Traversing of the data	Data elements can be accessed in one time (single run)	Traversing of data elements in one go is not possible
Ease of implementation	Simpler	Complex
Levels involved	Single	Multiple
Example	Array, queue, stack, linked list	Tree and Graph

Operations of Data Structures

- **Traversing** It means to access each data item exactly once so that it can be processed. For example, to print the names of all the students in a class.
- **Searching** It is used to find the location of one or more data items that satisfy the given constraint. Such a data item may or may not be present in the given collection of data items. For example, to find the names of all the students who secured 100 marks in mathematics.
- **Inserting** It is used to add new data items to the given list of data items. For example, to add the details of a new student who has recently joined the course.
- **Deleting** It means to remove (delete) a particular data item from the given collection of data items. For example, to delete the name of a student who has left the course.
- **Sorting** Data items can be arranged in some order like ascending order or descending order depending on the type of application. For example, arranging the names of students in a class in an alphabetical order, or calculating the top three winners by arranging the participants' scores in descending order and then extracting the top three.
- **Merging** Lists of two sorted data items can be combined to form a single list of sorted data items

1.1 ABSTRACT DATA TYPES (ADTS)

An abstract data type (ADT) is the way we look at a data structure, focusing on what it does and ignoring how it does its job

Data and its **corresponding operations** are concerned ignoring the implementation.

Advantages of ADT

- ADT is reusable and ensures robust data structure.
- It reduces coding efforts.

- Encapsulation ensures that data cannot be corrupted.
- ADT is based on principles of Object Oriented Programming (OOP) and Software Engineering (SE).
- It specifies error conditions associated with operations.

1.2 LIST ADT

- Insertion
- Deletion
- Traversal
- Searching
- Update
- Count_of_elements
- Merge

Linked Lists VS Array

- Both arrays and linked lists are a linear collection of data elements. But unlike an array, a linked list does not store its nodes in consecutive memory locations.
- Another point of difference between an array and a linked list is that a linked list does not allow random access of data. Nodes in a linked list can be accessed only in a sequential manner. But like an array, insertions and deletions can be done at any point in the list in a constant time.
- Another advantage of a linked list over an array is that we can add any number of elements in the list. This is not possible in case of an array. For example, if we declare an array as `int marks[20]`, then the array can store a maximum of 20 data elements only. There is no such restriction in case of a linked list.
- Thus, linked lists provide an efficient way of storing related data and performing basic operations such as insertion, deletion, and updation of information at the cost of extra space required for storing the address of next nodes.

1.3 ARRAY-BASED IMPLEMENTATION

An array is a collection of similar data elements. These data elements have the same data type. The elements of the array are stored in consecutive memory locations and are referenced by an index

For example,

```
int marks[10];
```

The above statement declares an array marks that contains 10 elements.

1 st element	2 nd element	3 rd element	4 th element	5 th element	6 th element	7 th element	8 th element	9 th element	10 th element
----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	----------------------------	-----------------------------

marks[0] marks[1] marks[2] marks[3] marks[4] marks[5] marks[6] marks[7] marks[8] marks[9]

Memory representation of array of 10 elements

Limitations of array

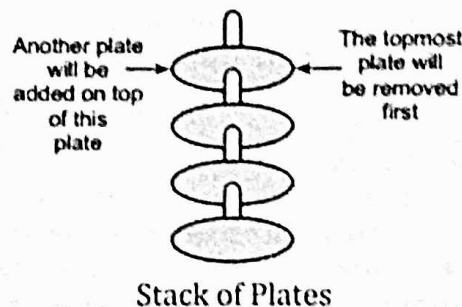
- Arrays are of fixed size. Unused allocated memory
- Data elements are stored in contiguous memory locations which may not be always available.
- Insertion and deletion of elements can be problematic because of shifting of elements from their positions

UNIT - 2 - LINEAR DATA STRUCTURES - STACK & QUEUE

2.1.STACK ADT

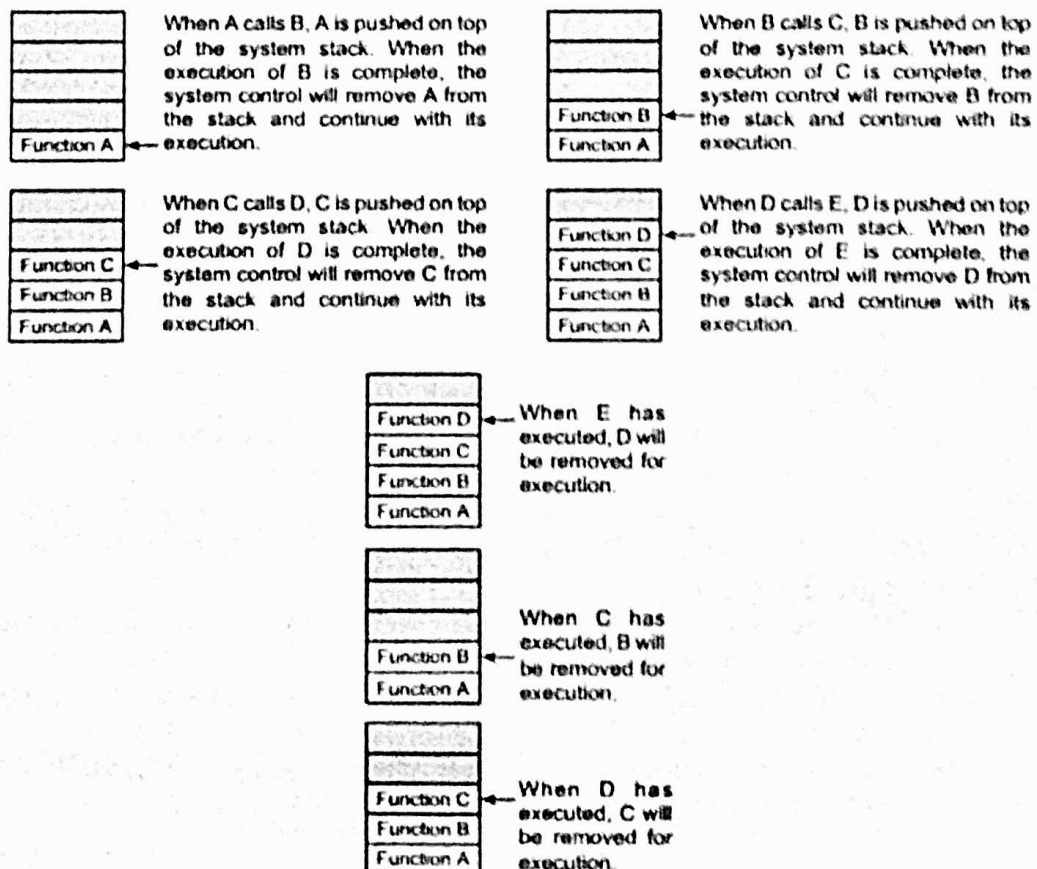
Introduction

- Stack is an ordered list of the same type of elements.
- It is a linear list where all insertions and deletions are permitted only at one end of the list.
- Stack is a LIFO (Last In First Out) structure.
- In a stack, when an element is added, it goes to the top of the stack. When a stack is completely full, it is said to be **Overflow state** and if stack is completely empty, it is said to be **Underflow state**



Need of Stack in Computer Science

Consider an example, where we are executing function A. In the course of its execution, function A calls another function B. Function B in turn calls another function C, which calls function D.



In order to keep track of the returning point of each active function, a special stack called system stack or call stack is used. Whenever a function calls another function, the

calling function is pushed onto the top of the stack. This is because after the called function gets executed, the control is passed back to the calling function

functions:

```

for all stack ∈ Stack, item ∈ element, max-stack-size ∈ positive integer
Stack CreateS(max-stack-size) ::=
    create an empty stack whose maximum size is max-stack-size
Boolean IsFull(stack, max-stack-size) ::=
    if (number of elements in stack == max-stack-size)
        return TRUE
    else return FALSE
Stack Add(stack, item) ::=
    if (IsFull(stack)) stack - full
    else insert item into top of stack and return
Boolean IsEmpty(stack) ::=
    if (stack == CreateS(max-stack-size))
        return TRUE
    else return FALSE
Element Delete(stack) ::=
    if (IsEmpty(stack)) return
    else remove and return the item on the top of the stack.
  
```

2.2 STACK – OPERATIONS

There are three basic operations performed in a Stack:

- Push() - is used to add or insert new elements into the stack.
- Pop() - is used to delete or remove an element from the stack.
- peek() - gets the top element of the stack without deleting it.

Other operations that can be performed in a stack are:

- isempty() - checks whether the stack is empty or not
- isnull() - used to check whether stack is full or not.
- size() - returns the number of elements in the stack.

Implementation of Stack using

- Arrays
- Linked List

2.2.1. Array representation of Stack

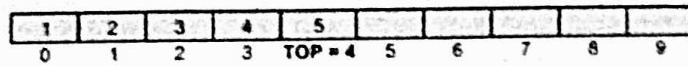
- Stacks can be represented as a linear array.
- Every stack has a variable called TOP associated with it, which is used to store the address of the topmost element of the stack. It is this position where the element will be added to or deleted from.
- There is another variable called MAX, which is used to store the maximum number of elements that the stack can hold.
- If TOP = NULL, then it indicates that the stack is empty and if TOP = MAX-1, then the stack is full.

A	AB	ABC	ABCD	ABCDE					
0	1	2	3	TOP = 4	5	6	7	8	9

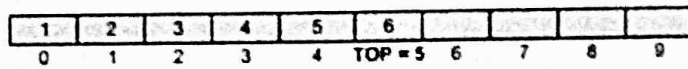
Push Operation

- The push operation is used to insert an element into the stack. The new element is added at the topmost position of the stack.
- Before inserting the value, we must first check if TOP=MAX-1, because if that is the case, then the stack is full and no more insertions can be done.
- If an attempt is made to insert a value in a stack that is already full, an OVERFLOW message is printed.

Eg: Consider the stack given in Fig



To insert an element with value 6, we first check if $TOP = MAX - 1$. If the condition is false, then we increment the value of TOP and store the new element at the position given by $stack[TOP]$. Thus, the updated stack becomes as shown in Fig



```

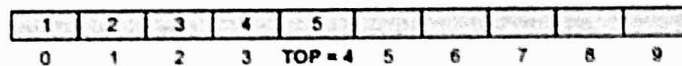
Step 1: IF TOP = MAX-1
        PRINT "OVERFLOW"
        Goto Step 4
    [END OF IF]
Step 2: SET TOP = TOP + 1
Step 3: SET STACK[TOP] = VALUE
Step 4: END
  
```

- In Step 1, we first check for the OVERFLOW condition.
- In Step 2, TOP is incremented so that it points to the next location in the array.
- In Step 3, the value is stored in the stack at the location pointed by TOP.

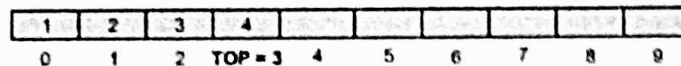
Pop operation

- The pop operation is used to delete the topmost element from the stack.
- Before deleting the value, we must first check if $TOP = NULL$ because if that is the case, then it means the stack is empty and no more deletions can be done.
- If an attempt is made to delete a value from a stack that is already empty, an UNDERFLOW message is printed.

Consider the stack given in Fig



To delete the topmost element, we first check if $TOP = NULL$. If the condition is false, then we decrement the value pointed by TOP. Thus, the updated stack becomes as shown in Fig



- In Step 1, we first check for the UNDERFLOW condition.
- In Step 2, the value of the location in the stack pointed by TOP is stored in VAL.
- In Step 3, TOP is decremented.

```

Step 1: IF TOP = NULL
        PRINT "UNDERFLOW"
        Goto Step 4
    [END OF IF]
Step 2: SET VAL = STACK[TOP]
Step 3: SET TOP = TOP - 1
Step 4: END
  
```

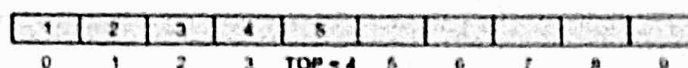
Peek Operation

Peek is an operation that returns the value of the topmost element of the stack without deleting it from the stack. However, the Peek operation first checks if the stack is empty, i.e., if $TOP = NULL$, then an appropriate message is printed, else the value is returned.

```

Step 1: IF TOP = NULL
        PRINT "STACK IS EMPTY"
        Goto Step 3
Step 2: RETURN STACK[TOP]
Step 3: END
  
```

Consider the stack given in Fig.

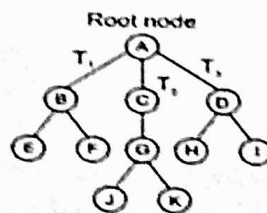


UNIT - 3 - NON-LINEAR DATA STRUCTURES - TREES

Topic No	Topic	Preparation status
3.1	Tree ADT	
3.2	Tree traversals	
3.3	Binary Tree ADT	
3.4	Expression trees	
3.5	Applications of trees	
3.6	Binary search tree ADT (set 2)	
3.7	Threaded Binary Trees	
3.8	AVL Trees	
3.9	B-Tree	
3.10	B+ Tree	
3.11	Heap(set 3)	
3.12	Applications of heap	

Tree - Introduction

A tree is recursively defined as a set of one or more nodes where one node is designated as the root of the tree and all the remaining nodes can be partitioned into non-empty sets each of which is a sub-tree of the root. Figure below shows a tree where node A is the root node; nodes B, C, and D are children of the root node and form sub-trees of the tree rooted at node A.



Basic Terminologies

Root node The root node R is the topmost node in the tree. If R = NULL, then it means the tree is empty.

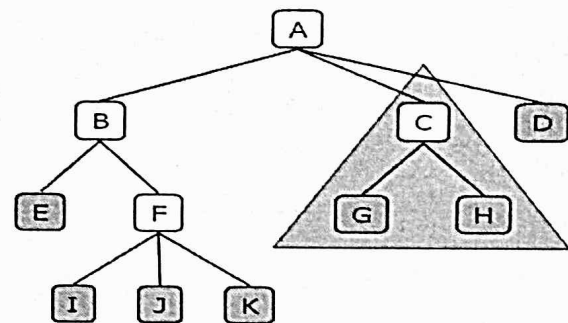
- **Sub-trees** If the root node R is not NULL, then the trees T1, T2, and T3 are called the sub-trees of R.
- **Leaf node** A node that has no children is called the leaf node or the terminal node. Path A sequence of consecutive edges is called a path. For example, in Fig. above, the path from the root node A to node I is given as: A, D, and I.
- **Ancestor node** An ancestor of a node is any predecessor node on the path from root to that node. The root node does not have any ancestors. In the tree given, nodes A, C, and G are the ancestors of node K.

- **Descendant node** A descendant node is any successor node on any path from the node to a leaf node. Leaf nodes do not have any descendants. In the tree given, nodes C, G, J, and K are the descendants of node A.
- **Level number** Every node in the tree is assigned a level number in such a way that the root node is at level 0, children of the root node are at level number 1. Thus, every node is at one level higher than its parent. So, all child nodes have a level number given by parent's level number + 1.
- **Degree** Degree of a node is equal to the number of children that a node has. The degree of a leaf node is zero.
- **In-degree** of a node is the number of edges arriving at that node.
- **Out-degree** of a node is the number of edges leaving that node

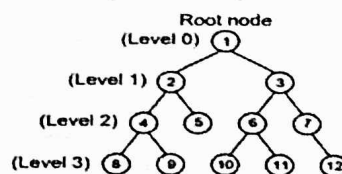
Illustration-2

- **Root** node without parent (A)
- **Siblings** nodes share the same parent
- **Internal node**: node with at least one child (A, B, C, F)
- **External node (leaf)**: node without children (E, I, J, K, G, H, D)
- **Ancestors** of a node: parent, grandparent, grand-grandparent, etc.
- **Descendant** of a node: child, grandchild, grand-grandchild, etc.
- **Depth** of a node: number of ancestors
- **Height** of a tree: maximum depth of any node (3)
- **Degree** of a node: the number of its children
- **Degree** of a tree: the maximum number of its node.

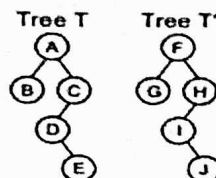
- **Subtree**: tree consisting of a node and its descendants



- **Parent** If N is any node in T that has left successor S1 and right successor S2, then N is called the parent of S1 and S2. Correspondingly, S1 and S2 are called the left child and the right child of N. Every node other than the root node has a parent.
- **Level number** Every node in the binary tree is assigned a level number. The root node is defined to be at level 0. The left and the right child of the root node have a level number 1. Similarly, every node is at one level higher than its parents. So all child nodes are defined to have level number as parent's level number + 1.



- **Degree of a node** It is equal to the number of children that a node has. The degree of a leaf node is zero. For example, in the tree, degree of node 4 is 2, degree of node 5 is zero and degree of node 7 is 1.
- **Sibling** All nodes that are at the same level and share the same parent are called siblings (brothers). For example, nodes 2 and 3; nodes 4 and 5; nodes 6 and 7; nodes 8 and 9; and nodes 10 and 11 are siblings
- **Leaf node** A node that has no children is called a leaf node or a terminal node. The leaf nodes in the tree are: 8, 9, 5, 10, 11, and 12.
- **Similar binary trees** Two binary trees T and T' are said to be similar if both these trees have the same structure. Figure shows two similar binary trees.

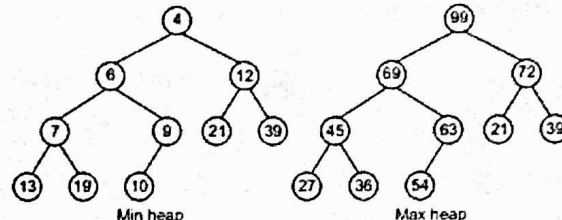


- **Copies** Two binary trees T and T' are said to be copies if they have similar structure and if they have same content at the corresponding nodes. Figure shows that T' is a copy of T.

UNIT – 3 –NON-LINEAR DATA STRUCTURES – TREES

3.11.HEAP

- A binary heap is a complete binary tree in which every node satisfies the heap property which states that:
- If B is a child of A, then $\text{key}(A) \geq \text{key}(B)$
- This implies that **elements at every node will be either greater than or equal to the element at its left and right child**. Thus, the root node has the highest key value in the heap. Such a heap is commonly known as a **max-heap**.



- Alternatively, **elements at every node will be either less than or equal to the element at its left and right child**. Thus, the root has the lowest key value. Such a heap is called a **min-heap**
- The properties of binary heaps are given as follows:
- Since a heap is defined as a complete binary tree, all its elements can be stored sequentially in an array.
- It follows the same rules as that of a complete binary tree. That is, **if an element is at position i in the array, then its left child is stored at position $2i$ and its right child at position $2i+1$. Conversely, an element at position i has its parent stored at position $i/2$**
 - Being a complete binary tree, all the levels of the tree except the last level are completely filled.
 - The height of a binary tree is given as $\log_2 n$, where n is the number of elements.
 - Heaps (also known as partially ordered trees) are a very popular data structure for implementing priority queues.

Advantages

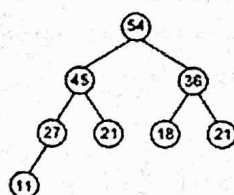
- A binary heap is a useful data structure in which **elements can be added randomly but only the element with the highest value is removed in case of max heap and lowest value in case of min heap**.
- A binary tree is an efficient data structure, but a **binary heap is more space efficient and simpler**.

Inserting a New Element in a Binary Heap

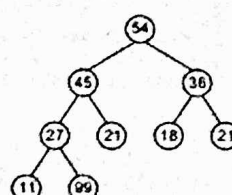
Consider a max heap H with n elements. Inserting a new value into the heap is done in the following two steps:

1. Add the new value at the bottom of H in such a way that H is still a complete binary tree but not necessarily a heap.
2. Let the new value rise to its appropriate place in H so that H now becomes a heap as well. To do this, compare the new value with its parent to check if they are in the correct order. If they are, then the procedure halts, else the new value and its parent's value are swapped and Step 2 is repeated.

Consider the max heap given in Fig. (a) and insert 99 in it



Binary heap(a)



(b) after inserting 99

UNIT - 4 -NON-LINEAR DATA STRUCTURES -

Topic No	Topic	Preparation status
4.1	Definition	
4.2	Representation of Graph	
4.3	Types of graph	
4.4	Breadth-first traversal	
4.5	Depth-first traversal	
4.6	Topological Sort	
4.7	Bi-connectivity	
4.8	Cut vertex	
4.9	Euler circuits	
4.10	Applications of graphs	

4.1 DEFINITION

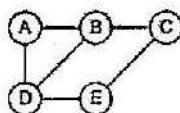
- A graph is an abstract data structure that is used to implement the mathematical concept of graphs.
- It is basically a collection of vertices (also called nodes) and edges that connect these vertices.
- A graph is often viewed as a generalization of the tree structure, where instead of having a purely parent-to-child relationship between tree nodes, any kind of complex relationship can exist.

Uses of Graphs

Graphs are widely used to model any situation where entities or things are related to each other in pairs. For example, the following information can be represented by graphs:

- Family trees in which the member nodes have an edge from parent to each of their children.
- Transportation networks in which nodes are airports, intersections, ports, etc. The edges can be airline flights, one-way roads, shipping routes, etc.

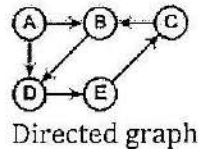
A graph G is defined as an ordered set (V, E) , where $V(G)$ represents the set of vertices and $E(G)$ represents the edges that connect these vertices.



Undirected graph

- A graph with $V(G) = \{A, B, C, D \text{ and } E\}$ and $E(G) = \{(A, B), (B, C), (A, D), (B, D), (D, E), (C, E)\}$. Note that there are five vertices or nodes and six edges in the graph.
- A graph can be directed or undirected. In an undirected graph, edges do not have any direction associated with them. That is, if an edge is drawn between nodes A and B, then the nodes can be traversed from A to B as well as from B to A.

- In a directed graph, edges form an ordered pair. If there is an edge from A to B, then there is a path from A to B but not from B to A. The edge (A, B) is said to initiate from node A (also known as initial node) and terminate at node B (terminal node).



4.1.1 Terminologies

Adjacent nodes or neighbours

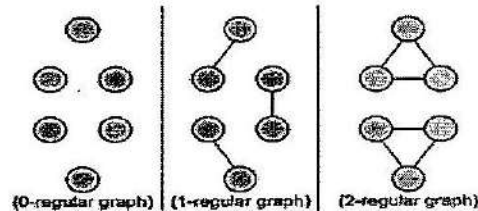
For every edge, $e = (u, v)$ that connects nodes u and v , the nodes u and v are the end-points and are said to be the adjacent nodes or neighbours.

Degree of a node

Degree of a node u , $\deg(u)$, is the total number of edges containing the node u . If $\deg(u) = 0$, it means that u does not belong to any edge and such a node is known as an isolated node.

Regular graph

It is a graph where each vertex has the same number of neighbours. That is, every node has the same degree. A regular graph with vertices of degree k is called a k -regular graph or a regular graph of degree k . Path A path P written as $P = \{v_0, v_1, v_2, \dots, v_n\}$, of length n from a node u to v is defined as a sequence of $(n+1)$ nodes. Here, $u = v_0$, $v = v_n$ and v_{i-1} is adjacent to v_i for $i = 1, 2, 3, \dots, n$.



Closed path

A path P is known as a closed path if the edge has the same end-points. That is, if $v_0 = v_n$. Simple path A path P is known as a simple path if all the nodes in the path are distinct with an exception that v_0 may be equal to v_n . If $v_0 = v_n$, then the path is called a closed simple path.

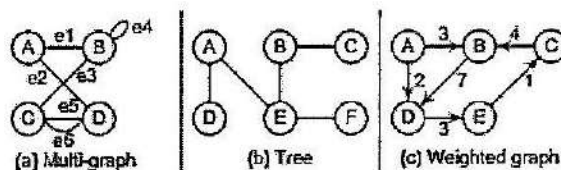
Cycle

A path in which the first and the last vertices are same.

A simple cycle has no repeated edges or vertices (except the first and last vertices).

Connected graph

A graph is said to be connected if for any two vertices (u, v) in V there is a path from u to v . That is to say that there are no isolated nodes in a connected graph. A connected graph that does not have any cycle is called a tree. Therefore, a tree is treated as a special graph



Complete graph

A graph G is said to be complete if all its nodes are fully connected. That is, there is a path from one node to every other node in the graph. A complete graph has $n(n-1)/2$ edges, where n is the number of nodes in G .

Clique

In an undirected graph $G = (V, E)$, clique is a subset of the vertex set $C \subseteq V$, such that for every two vertices in C , there is an edge that connects two vertices.

Labelled graph or weighted graph

A graph is said to be labelled if every edge in the graph is assigned some data. In a weighted graph, the edges of the graph are assigned some weight or length.

The weight of an edge denoted by $w(e)$ is a positive value which indicates the cost of traversing the edge.

Multiple edges

Distinct edges which connect the same end-points are called multiple edges. That is, $e = (u, v)$ and $e' = (u, v)$ are known as multiple edges of G .

Loop

An edge that has identical end-points is called a loop. That is, $e = (u, u)$.

Multi-graph

A graph with multiple edges and/or loops is called a multi-graph.

Size of a graph The size of a graph is the total number of edges in it.

4.1.2. Directed Graph

A directed graph G , also known as a digraph, is a graph in which every edge has a direction assigned to it. An edge of a directed graph is given as an ordered pair (u, v) of nodes in G .

For an edge (u, v) ,

- The edge begins at u and terminates at v .
- u is known as the origin or initial point of e . Correspondingly, v is known as the destination or terminal point of e .
- u is the predecessor of v . Correspondingly, v is the successor of u .
- Nodes u and v are adjacent to each other.

Out-degree of a node

The out-degree of a node u , written as $\text{outdeg}(u)$, is the number of edges that originate at u .

In-degree of a node

The in-degree of a node u , written as $\text{indeg}(u)$, is the number of edges that terminate at u .

Degree of a node

The degree of a node, written as $\text{deg}(u)$, is equal to the sum of in-degree and out-degree of that node. Therefore, $\text{deg}(u) = \text{indeg}(u) + \text{outdeg}(u)$.

Isolated vertex

A vertex with degree zero. Such a vertex is not an end-point of any edge. Pendant vertex (also known as leaf vertex) A vertex with degree one.

Cut vertex

A vertex which when deleted would disconnect the remaining graph. Source A node u is known as a source if it has a positive out-degree but a zero in-degree.

Sink

A node u is known as a sink if it has a positive in-degree but a zero out-degree. Reachability A node v is said to be reachable from node u , if and only if there exists a (directed) path from node u to node v . For example, if you consider the directed graph given in Fig. observe that node D is reachable from node A .

Strongly connected directed graph

A digraph is said to be strongly connected if and only if there exists a path between every pair of nodes in G . That is, if there is a path from node u to v , then there must be a path from node v to u .

Unilaterally connected graph

A digraph is said to be unilaterally connected if there exists a path between any pair of nodes u, v in G such that there is a path from u to v or a path from v to u , but not both.

Weakly connected digraph

A directed graph is said to be weakly connected if it is connected by ignoring the direction of edges. That is, in such a graph, it is possible to reach any node from any other node by traversing edges in any direction (may not be in the direction they point). The nodes in a weakly connected directed graph must have either out-degree or in-degree of at least 1.

Parallel/Multiple edges

Distinct edges which connect the same end-points are called multiple edges. That is, $e = (u, v)$ and $e' = (u, v)$ are known as multiple edges of G . In Fig. e_3 and e_5 are multiple edges connecting nodes C and D .

UNIT - 5 -SEARCHING, SORTING AND HASHING TECHNIQUES

Topic No	Topic	Preparation status
5.1	Searching	
5.2	Linear Search	
5.3	Binary Search	
5.4	Sorting	
5.5	Bubble sort	
5.6	Selection sort	
5.7	Insertion sort	
5.8	Shell sort	
5.9	Radix sort	
5.10	Hashing	
5.11	Hash Functions	
5.12	Separate Chaining	
5.12	Open Addressing	
5.13	Rehashing	
5.14	Extendible Hashing	

5.1. SEARCHING

- Searching means to find whether a particular value is present in an array or not.
- If the value is present in the array, then searching is said to be successful and the searching process gives the location of that value in the array. However, if the value is not present in the array, the searching process displays an appropriate message and in this case searching is said to be unsuccessful.
- There are two popular methods for searching the array elements: *linear search* and *binary search*.
- The algorithm that should be used depends entirely on how the values are organized in the array.
- For example, if the elements of the array are arranged in ascending order, then binary search should be used, as it is more efficient for sorted lists in terms of complexity

5.2. LINEAR SEARCH

- Linear search, also called as *sequential search*, is a very simple method used for searching an array for a particular value.
- It works by **comparing the value to be searched with every element of the array one by one in a sequence until a match is found.**
- Linear search is mostly used to search an **unordered list of elements** (array in which data elements are not sorted).

For example,

A[] is declared and initialized as,

int A[] = {10, 8, 2, 7, 3, 4, 9, 1, 6, 5};

if an array and the value to be searched is VAL = 7, then searching means to find whether the value '7' is present in the array or not. If yes, then it returns the position of its occurrence.

Here, POS = 3 (index starting from 0).

Algorithm

```

LINEAR_SEARCH(A, N, VAL)
Step 1: [INITIALIZE] SET POS = -1
Step 2: [INITIALIZE] SET I = 1
Step 3: Repeat Step 4 while I ≤ N
Step 4: IF A[I] = VAL
            SET POS = I
            PRINT POS
            Go to Step 6
        [END OF IF]
        SET I = I + 1
    [END OF LOOP]
Step 5: IF POS = -1
        PRINT "VALUE IS NOT PRESENT
        IN THE ARRAY"
    [END OF IF]
Step 6: EXIT
  
```

- In Steps 1 and 2 of the algorithm, we initialize the value of POS and I.
- In Step 3, a while loop is executed that would be executed till I is less than N (total number of elements in the array).
- In Step 4, a check is made to see if a match is found between the current array element and VAL. If a match is found, then the position of the array element is printed, else the value of I is incremented to match the next element with VAL.
- However, if all the array elements have been compared with VAL and no match is found, then it means that VAL is not present in the array

Complexity of Linear Search Algorithm

- Linear search executes in $O(n)$ time where n is the number of elements in the array.
- Obviously, the best case of linear search is when VAL is equal to the first element of the array. In this case, only one comparison will be made.
- Likewise, the worst case will happen when either VAL is not present in the array or it is equal to the last element of the array. In both the cases, n comparisons will have to be made.
- However, the performance of the linear search algorithm can be improved by using a sorted array

5.3. BINARY SEARCH

- Binary search is a searching algorithm that works efficiently with a sorted list.
- Take an analogy. How do we find words in a dictionary? We first open the dictionary somewhere in the middle. Then, we compare the first word on that page with the desired word whose meaning we are looking for. If the desired word comes before the word on the page, we look in the first half of the dictionary, else we look in the second half. Again, we open a page in the first half of the dictionary and compare the first word on that page with the desired word and repeat the same procedure until we finally get the word. The same mechanism is applied in the binary search.

Now, let us consider how this mechanism is applied to search for a value in a sorted array.

Consider an array A[] that is declared and initialized as

int A[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}; and the value to be searched is VAL = 9. The algorithm will proceed in the following manner.

BEG = 0, END = 10, MID = $(0 + 10)/2 = 5$

Now, VAL = 9 and A[MID] = A[5] = 5

A[5] is less than VAL, therefore, we now search for the value in the second half of the array. So, we change the values of BEG and MID.

Now, $BEG = MID + 1 = 6$, $END = 10$, $MID = (6 + 10)/2 = 16/2 = 8$
 $VAL = 9$ and $A[MID] = A[8] = 8$

$A[8]$ is less than VAL , therefore, we now search for the value in the second half of the segment.
 So, again we change the values of BEG and MID .

Now, $BEG = MID + 1 = 9$, $END = 10$, $MID = (9 + 10)/2 = 9$

Now, $VAL = 9$ and $A[MID] = 9$.

- In this algorithm, we see that BEG and END are the beginning and ending positions of the segment that we are looking to search for the element. MID is calculated as $(BEG + END)/2$. Initially, $BEG = \text{lower_bound}$ and $END = \text{upper_bound}$.
- The algorithm will terminate when $A[MID] = VAL$. When the algorithm ends, we will set $POS = MID$. POS is the position at which the value is present in the array.
- However, if VAL is not equal to $A[MID]$, then the values of BEG , END , and MID will be changed depending on whether VAL is smaller or greater than $A[MID]$.

(a) If $VAL < A[MID]$, then VAL will be present in the left segment of the array. So, the value of END will be changed as $END = MID - 1$.

(b) If $VAL > A[MID]$, then VAL will be present in the right segment of the array. So, the value of BEG will be changed as $BEG = MID + 1$.

Finally, if VAL is not present in the array, then eventually, END will be less than BEG . When this happens, the algorithm will terminate and the search will be unsuccessful.

Algorithm

BINARY_SEARCH(A, lower_bound, upper_bound, VAL)

Step 1: [INITIALIZE] SET $BEG = \text{lower_bound}$

$END = \text{upper_bound}$, $POS = -1$

Step 2: Repeat Steps 3 and 4 while $BEG \leq END$

Step 3: SET $MID = (BEG + END)/2$

Step 4: IF $A[MID] = VAL$

SET $POS = MID$

PRINT POS

Go to Step 6

ELSE IF $A[MID] > VAL$

SET $END = MID - 1$

ELSE

SET $BEG = MID + 1$

[END OF IF]

[END OF LOOP]

Step 5: IF $POS = -1$

PRINT "VALUE IS NOT PRESENT IN THE ARRAY"

[END OF IF]

Step 6: EXIT

Complexity of Binary Search Algorithm

- The complexity of the binary search algorithm can be expressed as $f(n)$, where n is the number of elements in the array.
- The complexity of the algorithm is calculated depending on the number of comparisons that are made.
- In the binary search algorithm, we see that with each comparison, the size of the segment where search has to be made is reduced to half. Thus, we can say that, in order to locate a particular value in the array, the total number of comparisons that will be made is given as $2^{f(n)} > n$ or $f(n) = \log_2 n$

5.4 SORTING

Sorting means arranging the elements of an array so that they are placed in some relevant order which may be either ascending or descending.

Unit - I
Linear Data Structures - LIST

Topic

Page No.

1.1 Abstract Data Types (ADTs)

I.3

1.2 List ADT

I.3

1.3 Array based Implementation

I.6

1.4 Linked list Implementation

I.10

1.5 Singly linked list

I.12

1.6 Circularly linked list

I.17

1.7 Doubly linked list

I.19

1.8 Applications of lists

I.22

1.8.1 Polynomial Manipulation

I.22

Introduction

Data structure is a representation of the logical relationship existing between individual elements of data. A data structure is a particular way of organizing data in a computer so that it can be used effectively.

- Data structure specifies
 - organizing of data
 - accessing methods
 - degree of associativity
 - processing alternatives for information.

Characteristics of a Data structure

- Correctness - DS should implement its interface correctly.
- Time Complexity - running time / the execution time of operations of data structures must be small as possible.
- Space Complexity - memory usage of a DS operation should be less.

Execution Time Cases

- Worst Case - a particular DS operation takes maximum time it can take.
- Average case - depicting the average execution time of an operation of a DS. If an operation takes $j(n)$ time in execution, then m operations will make $m \cdot j(n)$ time.
- Best Case - depicting the least possible execution time of an operation of a DS.

Classification of Data Structure

1. Primitive - Integer, Floating point, character, Pointer
2. Non Primitive -
 - Array
 - List
 - Linear list
 - Stack
 - Queue
 - Non Linear list
 - Graph
 - Tree
 - File

Data structures are broadly classified into 2

- primitive data structure
- non-primitive data structure

Primitive data structure

- are basic structures & are directly operated upon by machine instructions.

- have different representation on different computers.
- eg: integer, float, character, pointers

Nonprimitive data structure

- derived from primitive data structure.
- emphasize on structuring of a group of homogeneous or heterogeneous data items.
- eg: array, list, file.
- categories
 1. linear data structure
 2. non-linear data structure

Linear data structure

- elements are connected in linear fashion by means of logical or in sequential memory location.
- There are 2 ways to represent a linear data structure
 - static memory location
 - dynamic memory location
- possible operations on the linear data structure are traversal, insertion, deletion, searching, sorting & merging.
- eg: stack & Queue.

Non-linear data structure

- data items are not arranged in a sequence.
- eg: tree, graph.

Difference between Linear & Non Linear Data Structure

Linear Data Structure

- every item is related to its previous & next item
- data is arranged in linear sequence
- data items can be traversed in a single run.
- implementation is easy
- eg: array, queue, list

Non Linear Data Structure

- Every item is attached with many other items.
- data is not arranged in sequence.
- data cannot be traversed in a single run.
- implementation is difficult.
- eg: tree, graph.

1.1 Abstract Data Type (ADT)

An ADT is the specification of a datatype within some language, independent of an implementation.

- The interface for the ADT is defined in terms of a type & a set of operations on that type. The behavior of each operation is determined by its inputs & outputs.
- An ADT ~~also~~ does not specify how the datatype is implemented. These implementation details are hidden from the user of the ADT & protected from the outside access.

1.2 List ADT

- A list contains elements of same type arranged in sequential order & following operations can be performed on the list.
- get() - return an element from the list at any given position
- insert() - insert an element at any position of the list.
- remove() - remove the first occurrence of any element from a non-empty list.
- removeAt() - remove the element at a specified location from a non-empty list.

Unit-II

Linear Data Structures - Stacks, Queues

Topic

Page No.

2.1 Stack ADT

2.2 Operations

2.3 Applications

2.4 Evaluating Arithmetic Expressions

2.5 Conversion of Infix to Postfix Expression

2.6 Queue ADT

2.7 Operations

2.8 Circular Queue

2.9 Priority Queue

2.10 DeQueue

2.11 Applications of Queue

II. 1

II. 2

II. 5

II. 6

II. 6

II. 8

II. 9

II. 11

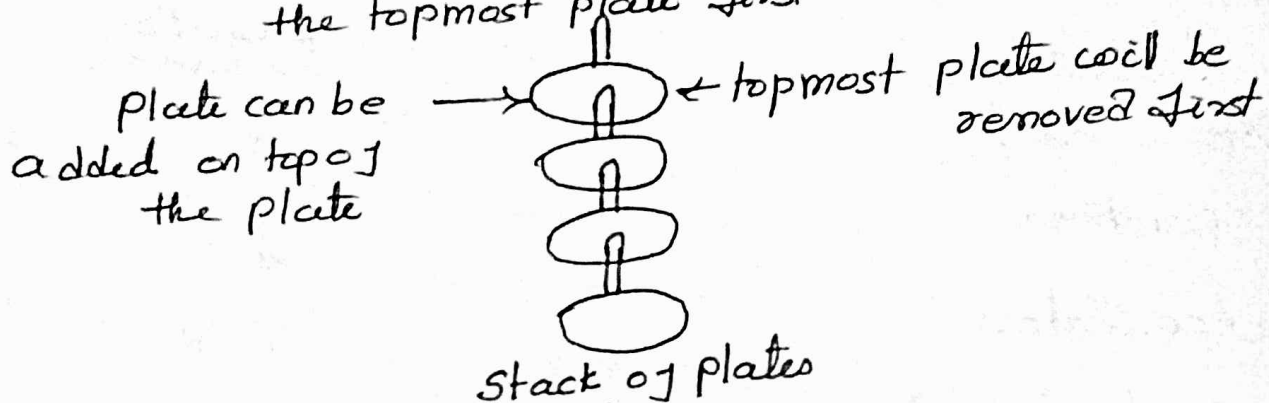
II. 15

II. 12

II. 16

Introduction

- stack is an important data structure that store its elements in an ordered manner.
- Analogy - pile of plates where one plate is place on top of another
 - if we want to remove a plate, we can remove the topmost plate first.



- A stack is a linear data structure which uses same principle, i.e., the elements in a stack are added & removed only from one end, which is called the top.
- Hence, a stack is called a LIFO (Last-in-First-out) data structure as the element that was inserted last is the first one to be taken out.
- stack is CS - junction calls.

Fun. C
Fun. D
Fun. C
Fun. B
Function A

2.1 Stack ADT

2 basic operations performed in a stack

push() - to add/insert new elements into the stack

pop() - to delete/remove an element from the stack.

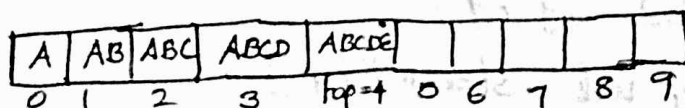
Other operations that can be performed in a stack are

peek() - get the top element of the stack without deleting it.

isEmpty() - checks whether the stack is empty/not
isFull() - used to check whether stack is full/not
size() - returns the number of elements in the stack.

Array representation of stack

- in the computer's memory, stack can be represented as a linear array.
- every stack has a variable called top associated with it, which is used to store the address of the topmost element of the stack.
- in this position where the element can be added/deleted from it.
- max - store the maximum no. of elements that the stack can hold.
- If $\text{top} = \text{null}$, then it indicates the stack is empty & if $\text{top} = \text{max} - 1$, then the stack is full.



2.2

Operations on a stack

Push operations

- push operation is used to insert an element into the stack.
- new element is added at the topmost position of the stack.
- before inserting the value, we must first check if $\text{top} = \text{max} - 1$
- overflow message - if stack is already full.

step 1: If $\text{top} = \text{max} - 1$,
 print 'overflow'
 goto step 4

2: set $\text{top} = \text{top} + 1$

3: set $\text{stack}[\text{top}] = \text{value}$

4: End

Unit-III

Non-linear Data Structures - Trees

<u>Topic</u>	<u>Page No.</u>
	<u>III. 1</u>
3.1 Tree ADT	<u>III. 6</u>
3.2 Tree Traversals	<u>III. 3</u>
3.3 Binary Tree ADT	<u>III. 5</u>
3.4 Expression Trees	<u>III. 8</u>
3.5 Application of trees	<u>III. 9</u>
3.6 Binary Search tree ADT	<u>III. 12</u>
3.7 Threaded Binary tree	<u>III. 13</u>
3.8 AVL trees	<u>III. 15</u>
3.9 B-Tree	<u>III. 16</u>
3.10 B+ Tree	<u>III. 17</u>
3.11 Heap	<u>III. 18</u>
3.12 Applications of Heap	

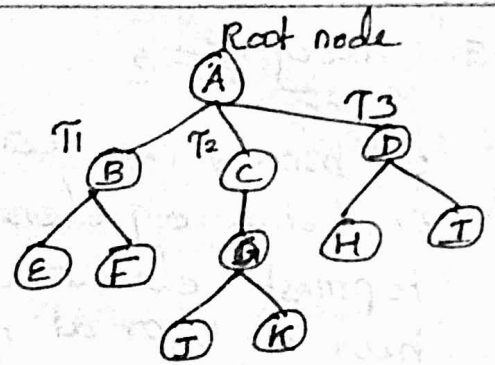
Introduction

A tree is recursively defined as a set of nodes where one node is designated as the root of the tree & all the remaining nodes can be partitioned into non-empty sets each of which is a sub-tree of the root.

1. Root node - R is the topmost node in the tree. If $R = \text{null}$, then it means the tree is empty.
2. Subtree - If the root node R is not null, then the trees T_1, T_2 & T_3 are sub-trees of R.
3. Leaf node - A node that has no children is called leaf node or the terminal node.
4. Path - A sequence of consecutive edges is called a path.
5. Ancessor node - An ancestor of a node is any predecessor node on the path from root to that node. Root node does not have any ancestors.
6. Descendant node - A descendant node is any successor node on any path from the node to a leaf node.
7. Level number - Every node in the tree is assigned a level number in such a way that the root node is at level 0, children of the root are at level number 1.
8. Degree - Degree of a node is equal to the number of children that the node has.

Types of trees

1. General trees
2. Forests
3. Binary trees
4. Binary search tree
5. Expression trees
6. Tournament trees.

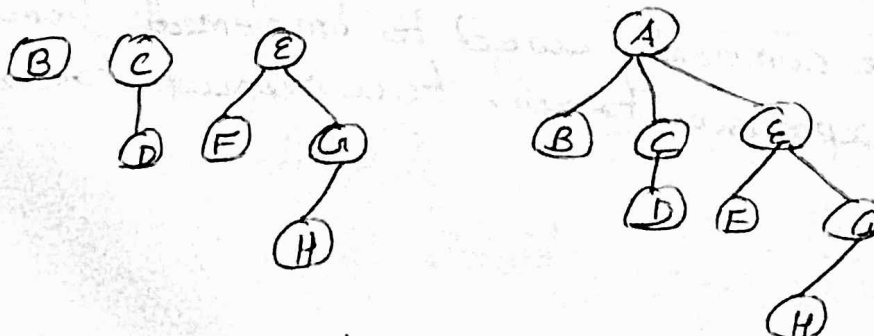


General trees

- are data structures that store elements hierarchically.
- the top node of a tree is the root node & each node, except the root, has a parent.
- A node in a general (except leaf) may have zero or more sub-tree.
- A general tree with 3 sub-trees per node are called ternary tree.

Forests

- A forest is a disjoint union of trees.
- A set of disjoint trees is obtained by deleting the root & edges connecting the root node to the nodes at level 1.

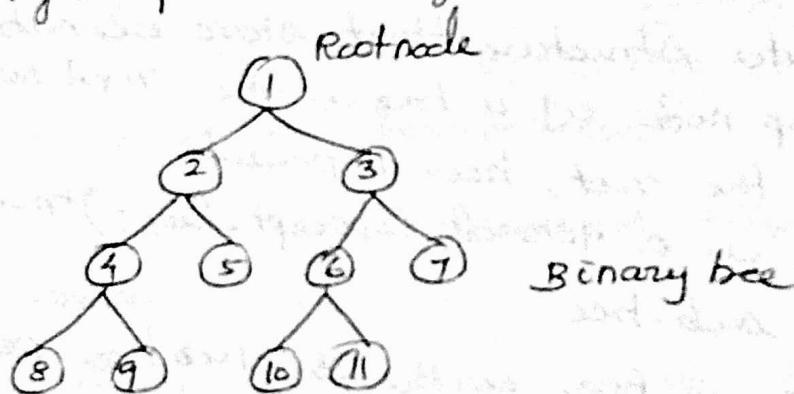


Forest & its corresponding tree

3.3 Binary Trees

A binary tree is a data structure that is defined as a collection of elements called nodes. In a binary tree, the topmost element is called the root node, & each node has 0, 1, or at most 2 children.

- every node contains
 - a data element
 - left pointer to left child
 - right pointer to right child.



In-degree / Out-degree of a node

- it is the number of edges arriving at a node. The root node is the only node that has an in-degree equal to zero.
- the out-degree of a node is the number of edges leaving that node.

Binary trees are commonly used to implement binary search trees, expression trees, tournament trees, binary heaps.

KINGS COLLEGE OF ENGINEERING

ASSESSMENT TEST – I (JULY 2019)

CS8391 DATA STRUCTURES

Class : II CSE

Date: 24.7.19

Maximum Marks : 50

Time: 1 1/2 Hrs

Answer all the questions

PART - A (10 * 2 = 20)

1. Make a comparison between a linked list and a linear array. Which one will you prefer to use and when?.
2. Distinguish between linear and non-linear data structure.
3. Why is a doubly linked list more useful than a singly linked list?.
4. What are Priority queues?. What are the ways to implement priority queues?.
5. Convert $3*3/(4-1)+6*2$ expression into postfix form.

PART – B (13 * 2 = 26)

- 6.a.i. State the polynomial representation for $6x^3+9x^2+7x+1$ using linked list. (6)
- ii. Write procedure to add two polynomial using linked list and explain with suitable eg. (7)
- (or)
- 6.b. Develop an algorithm to perform the following operations in a doubly linked list (13)
 - i. Insert a node at the end of the list. (5)
 - ii. Delete the last node in the list. (4)
 - iii. Search an element in the list. (4)
- 7.a.i. Write the procedure to convert the infix expression to postfix expression and steps involved in evaluating the postfix expression. (6)
- ii. Convert the expression $A-(B/C+(D\%E\cdot F)/G)\cdot H$ to postfix form. Evaluate the given postfix expression $9\ 3\ 4\ *\ 8\ +\ 4\ /\ -$ (7)
- (or)
- 7.b.i. Write the procedure to insert and delete an element from a circular queue. (7)
- ii. State the limitations of Linear queue with a suitable illustration. Mention the applications of circular queue. (6)

PART - C (14 * 1 = 14)

- 8.a.i. Write a program that removes all nodes that have duplicate information . (10)
ii. Write the procedure to swap 2 adjacent elements by adjusting only pointers using singly linked list. (4)

(or)

- 8.b.i. Consider the queue given below which has Front =1 and Rear=5. Perform the following operations on the queue and write the procedure. (10)

Add F

Add G

Delete rear 2 elements

	A	B	C	D	E				
--	---	---	---	---	---	--	--	--	--

- ii. Given 2 sorted lists L1, L2. Write a procedure to compute L1 intersect L2. (4)

KINGS COLLEGE OF ENGINEERING
ASSESSMENT TEST – II (SEP 2019)
CS8391 DATA STRUCTURES

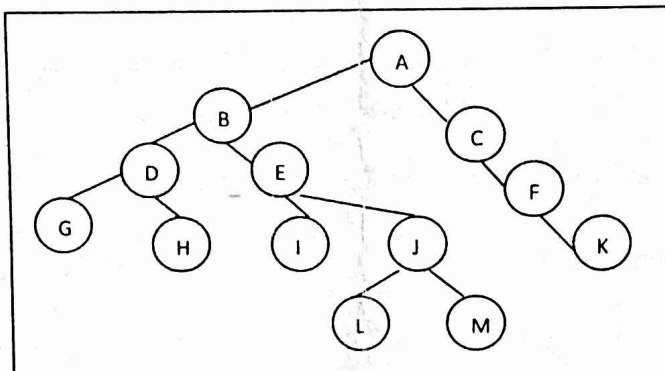
Class : II CSE
Maximum Marks : 100

Date: 06.9.19
Time: 9.30 am-12.30 pm

Answer all the questions

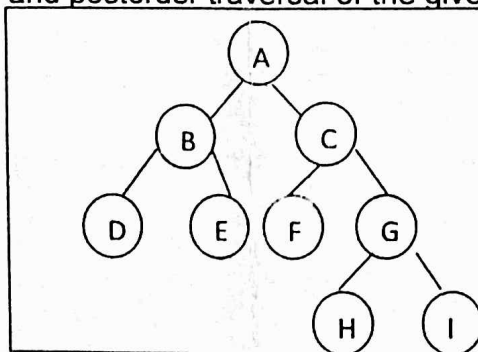
PART - A (10 x 2 = 20)

1. Write short note on extended binary tree and full binary tree. Give example.
2. Enumerate on the applications of Circular queue.
3. What is the maximum height of the tree with 32 nodes?
4. From the given figure



Find height of the tree, depth of the node F, internal nodes of the tree

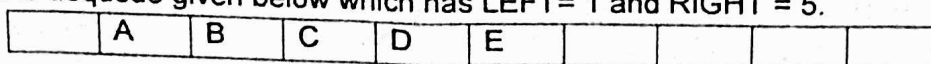
5. Find the inorder, preorder and postorder traversal of the given tree.



6. Define Euler's Circuit.
7. List the application of trees.
8. Mention the limitations of adjacency matrix
9. What are AVL trees?.
10. Write short note on Complete binary tree.

PART – B (5 *13 = 65)

11. a.i. Develop an algorithm to perform various operations on Circular Queue. How it is better than a linear queue?.
- ii. Consider the dequeue given below which has LEFT= 1 and RIGHT = 5.



Now perform the following operations on the queue: Add F on the left, Add G on the right, Add H on the right, Delete two letters from left, Add I on the right, Add J on the left, Delete two letters from right. Discuss briefly about dequeue variants. (6)

(or)

11.b.i. Elaborate on the Priority Queue implementation using Linked representation and array representation. (9)

ii. Propose a solution that lies a tradeoff between the frequency of overflow and space allocated in Queues. (4)

12.a.i. Write an algorithm to perform preorder, inorder and postorder traversal on a binary tree. Give suitable illustration (7)

ii. "Threaded binary trees are called efficient trees". Justify with suitable illustration. (6)

(or)

12.b. Appraise the various methods in which a tree can be represented. Discuss the advantage and disadvantage of each method. (13)

13.a.i. Construct the max heap for the following 90, 150, 70, 40, 100, 20, 30, 10, 110. Show the result of delete max. (7)

ii. Identify the steps to convert the general tree into a binary tree. Give example. (6)

(or)

13.b. Develop procedure for Insertion, Deletion operation on Binary Heap. Explain with suitable example. (13)

14.a.i. Demonstrate creating a B tree of order 5 by inserting the following elements: 3, 14, 7, 1, 8, 5, 11, 17, 13, 6, 23, 12, 20, 26, 4, 16, 24, 25, 19. (6)

ii. Explain the procedure for insertion and deletion of elements in B Tree. (7)

(or)

14.b.i. With suitable illustration infer insertion and deletion operation on B+ tree. Distinguish between B Tree and B+ tree (13)

15.a. Construct a Binary Search tree with the input given below (6)

98, 2, 48, 12, 56, 32, 4, 67, 23, 87, 55, 46. Insert 21, 39, 45, 54 and 63 into the tree. (4)

Delete values 23, 56, 2, 45 from the tree. (3)

(or)

15.b. Distinguish BFS and DFS operations on a graph. Write the procedure and Illustrate. (13)

PART - C (1 * 14 = 14)

16.a.i. Create a binary search tree with the input given below.

98, 2, 48, 12, 56, 32, 4, 67, 23, 87, 23, 55, 46. Insert 21, 39, 45, 54, and 63 into the tree

Delete values 23, 56, 2, and 45 from the tree. Write procedure for operations.

ii. Construct an AVL tree by inserting the following elements in the given order: 63, 9, 19, 27, 18, 108, 99, 81. How is a AVL tree better than binary search tree?

(or)

16.b.i. Develop an algorithm to perform various operations on AVL trees. Give illustration.

ii. Enumerate on how to balance AVL trees. Give example.

KEY FOR EVALUATION
ASSESSMENT TEST – 2 (SEP 2019)
CS8391 DATA STRUCTURES

PART - A (10 x 2 = 20)

1. Write short note on extended binary tree and full binary tree. Give example.
 If each node of binary tree has either two children or no child at all, is said to be a **Full Binary Tree**. Full binary tree is also called as **Strictly Binary Tree**.



Fig. Full Binary Tree

Extended binary tree consists of replacing every null subtree of the original tree with special nodes

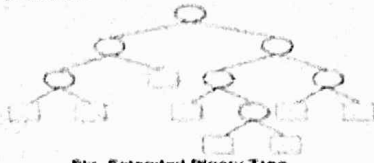
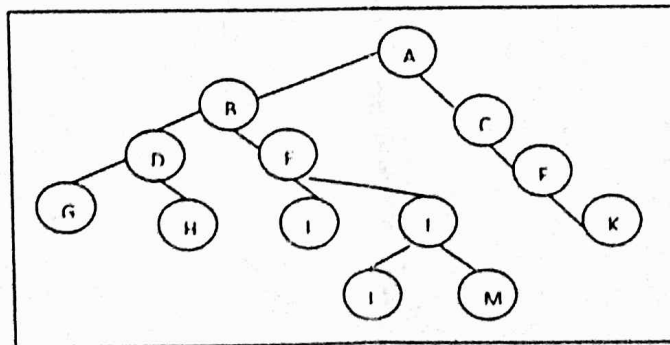


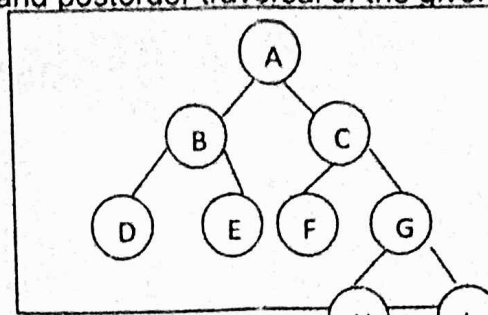
Fig. Extended Binary Tree

2. Enumerate on the applications of Circular queue.
 Memory Management: The unused memory locations in the case of ordinary queues can be utilized in circular queues.
 Traffic system: In computer controlled traffic system, circular queues are used to switch on the traffic lights one by one repeatedly as per the time set.
 CPU Scheduling: Operating systems often maintain a queue of processes that are ready to execute or that are waiting for a particular event to occur.
3. What is the maximum height of the tree with 32 nodes?
 Binary tree – n nodes – $n-1$ height
 Poorly balanced tree – 32 levels
4. From the given figure



Find height of the tree, depth of the node F, internal nodes of the tree
 Height of the tree(longest path edges)-4
 Depth of the node F(edges to the root)-2
 Internal nodes-B,D,E,J,C,F,A

5. Find the inorder, preorder and postorder traversal of the given tree.



Inorder (L Root R)-D B E A F C H G I
 Preorder(Root L R)-A B D E C F G H I
 Postorder(L R Root)-D E B F H I G C A

6. Define Euler's Circuit.

Eulerian Path is a path in graph that visits every edge exactly once. Eulerian Circuit is an Eulerian Path which starts and ends on the same vertex. A graph is called Eulerian if it has an Eulerian Cycle and called Semi-Eulerian if it has an Eulerian Path.

7. List the application of trees.

- Trees are used to store simple as well as complex data. Here simple means an integer value character value and complex data means a structure or a record.
- Trees are often used for implementing other types of data structures like hash tables, sets, and maps.
- Trees are an important data structure used for compiler construction.
- Trees are also used in database design.
- Trees are used in file system directories.
- Trees are also widely used for information storage and retrieval in symbol tables (any 4)

8. Mention the limitations of adjacency matrix

Limitations of Adjacency Matrix

- It consumes huge amount of memory for storing big graphs.
- It requires huge efforts for adding or removing a vertex. If you are constructing a graph in dynamic structure, adjacency matrix is quite slow for big graphs

9. What are AVL trees?

- AVL tree is a height balanced tree.
- It is a self-balancing binary search tree.
- AVL tree is another balanced binary search tree.
- It was invented by Adelson-Velskii and Landis.
- AVL trees have a faster retrieval.
- It takes $O(\log n)$ time for addition and deletion operation.
- In AVL tree, heights of left and right subtree cannot be more than one for all nodes (any 4)

10. Write short note on Complete binary tree.

- If all levels of tree are completely filled except the last level and the last level has all keys as left as possible, is said to be a **Complete Binary Tree**.
- Complete binary tree is also called as **Perfect Binary Tree**.

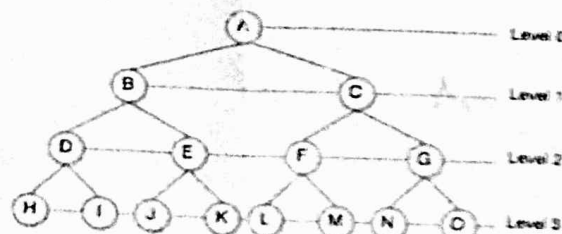


Fig. Complete Binary Tree

PART – B (5 *13 = 65)

11. a.i. Develop an algorithm to perform various operations on Circular Queue. How it is better than a linear queue?.

(7)

Insertion, Deletion operation – 6

Memory utilization efficiency-1

- ii. Consider the dequeue given below which has LEFT= 1 and RIGHT = 5.

	A	B	C	D	E				
--	---	---	---	---	---	--	--	--	--

Now perform the following operations on the queue: Add F on the left, Add G on the right, Add H on the right, Delete two letters from left, Add I on the right, Add J on the left, Delete two letters from right. Discuss briefly about dequeue variants.

(6)

F	A	B	C	D	E	G	H		
		B	C	D	E	G	H		
		B	C	D	E	G	H	I	
	J	B	C	D	E	G	H	I	
	J	B	C	D	E	G			

Procedure -3

Fig-3

(or)

- 11.b.i. Elaborate on the Priority Queue implementation using Linked representation and array representation.

(9)

Procedure-5+4

- ii. Propose a solution that lies a tradeoff between the frequency of overflow and space allocated in Queues.

(4)

Multiple queues – Q A grow from L to R while queue B from right to left

- 12.a.i. Write an algorithm to perform preorder, inorder and postorder traversal on a binary tree. Give suitable illustration

(7)

Inorder – LROOT R

Preorder- ROOT LR

Postorder- LR ROOT

Algorithm & eg – 2 +1 for fig.

- ii. "Threaded binary trees are called efficient trees". Justify with suitable illustration.

(6)

NULL pointer – replaced by threads

1 way thread – left or right threaded binary tree (3)

2 way thread – thread on both sides(3)

Search efficiency

eg

(or)

- 12.b. Appraise the various methods in which a tree can be represented . Discuss the advantage and disadvantage of each method.

(13)

General Tree, Forests, Binary, Binary search trees, Expression tree, Tournament tree, B tree, B+ tree.

- 13.a.i. Construct the max heap for the following 90, 150, 70, 40, 100, 20, 30, 10, 110. Show the result of delete max.

(7)

Insertion of elements and deletion (5 marks for insertion, 2 for deletion)

ii. Identify the steps to convert the general tree into a binary tree. Give example. (6)

root=root

left=leftmost

right=right sibling

7 steps for building tree

(or)

13.b. Develop procedure for Insertion, Deletion operation on Binary Heap. Explain with suitable example. (13)

Procedure with illustration- maintaining heap property (max / min)

14.a.i. Demonstrate creating a B tree of order 5 by inserting the following elements: 3,14,7,1,8,5,11,17,13,6,23,12,20,26,4,16,24,25,19. (6)

B tree – insertion-6

ii .Explain the procedure for insertion and deletion of elements in B Tree. (7)

procedure for insertion(4), deletion(3)

(or)

14.b.i.With suitable illustration infer insertion and deletion operation on B+ tree. Distinguish between B Tree and B+ tree(13)

Procedure for insertion, deletion (8)

6 points (6)

15.a. Construct a Binary Search tree with the input given below (6)

98, 2, 48, 12, 56, 32, 4, 67, 23, 87, 55, 46. Insert 21, 39, 45, 54 and 63 into the tree. (4)

Delete values 23, 56, 2, 45 from the tree. (3)

(or)

15.b. Distinguish BFS and DFS operations on a graph. Write the procedure and Illustrate . (13)

BFS – root and explore all neighbours

DFS-root, deeper and deeper

Procedure & eg (7+6)

PART – C (1 * 14 = 14)

16.a.iCreate a binary search tree with the input given below.

98, 2, 48, 12, 56, 32, 4, 67, 23, 87, 23, 55, 46. Insert 21, 39, 45, 54, and 63 into the tree

Delete values 23, 56, 2, and 45 from the tree. Write procedure for operations.

Insertion procedure with eg.

Deletion procedure with eg.

ii. Construct an AVL tree by inserting the following elements in the given order. (6)

63, 9, 19, 27, 18, 108, 99, 81.How is a AVL tree better than binary search tree?.

AVL trees construction

(or)

16.b.i.Develop an algorithm to perform various operations on AVL trees. Give illustration.

ii.Enumerate on how to balance AVL trees. Give example.

Insertion procedure with eg.

Deletion procedure with eg.

AVL trees – rotation (4 types – LL, RR, LR, RL)

28/44

✓
KINGS COLLEGE OF ENGINEERING
MODEL EXAMINATION (OCT' 2019)
CS8391 DATA STRUCTURES

Class : II CSE
Maximum Marks : 100

Date: 12.10.19
Time: 9.30 am-12.30 pm

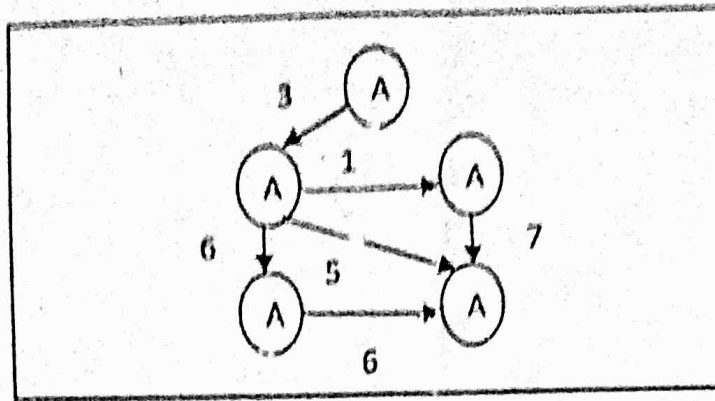
Answer all the questions

PART - A (10 x 2 = 20)

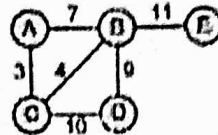
1. State the advantage of ADT
2. Distinguish linear data structure with non-linear data structure.
3. Tabulate the operations that can be performed on queue
4. Convert $3*3/(4-1)+6*2$ expression into postfix form.
5. Explain height of node & tree, depth of node, degree of node and edge of a tree.
6. Enumerate on Binary tree properties.
7. Write short note on Adjacency list graph representation?
8. When is an undirected graph called Biconnected?
9. What is hashing?
10. Give the advantage and disadvantage of separate chaining over linear probing.

PART - B (5 *13 = 65)

- 11.a.i. State the polynomial representation for $6x^3+9x^2+7x+1$ using linked list. Write procedure to add and multiply two polynomial and explain with suitable example. (7)
ii. What are the ways to insert a node in linked list?. Write an algorithm for inserting a node before a given node in a linked list. (6)
(or)
- 11.b.i. What are the various operations on array?. Write a procedure to insert an element in the middle of the array. (7)
ii. Write a procedure to delete the last node from a circular linked list. (6)
- 12.a.i. Write procedure and explain linked list representation of Queue. Give procedure for conversion of Infix to Prefix expression. (13)
(or)
- 12.b. Describe the procedure for array and linked list implementation of Priority queue. Give the application of Queue & Dequeue. (13)
- 13.a.i. Write a routine for AVL tree insertion. Insert the following elements in the empty tree and how do you balance the tree after each element insertion?. 2, 5, 4, 6, 7, 9, 8, 3, 1, 10. Brief about B+ tree. Discuss the application of heap. (13)
(or)
- 13.b.i. Give algorithm and illustrate Inorder, Preorder and Postorder traversal of a tree. Write the routine for deletion of a node in Binary search tree. Give eg. (13)
- 14.a.i. Apply an appropriate algorithm to find the shortest path from A to every other node of A, for the given graph below. Write the procedure. (7)



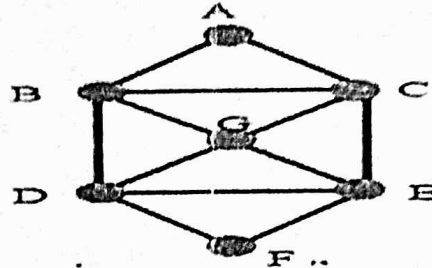
a.ii. Construct the MST of the graph given below and write the procedure. (8)



(or)

14.b.i. Illustrate with example Strongly Connected components. (6)

ii. Find an Euler path and circuit using DFS for the following graph. (7)



15.a.i. Distinguish between linear search and binary search. State and explain the algorithms for both the search with example. (7)

ii. Sort the sequence of numbers 3, 1, 4, 1, 5, 9, 2, 6, 5 using Insertion sort. Explain the routine and its performance. (6)

(or)

15.b.i. Sort the given integers and explain the intermediate results using shell sort. (7)

35, 12, 14, 9, 15, 45, 32, 95, 40, 5. Explain the algorithm. Mention its complexity, pros & cons. (7)

ii. Explain Rehashing and extendible hashing. (6)

PART - C (1 * 15 = 15)

16.a. Write procedure to

Compute intersect of 2 lists (5)

Reverse a list using non recursive procedure (5)

Duplicate element in a list (5)

(or)

16.b. Create a B tree of order 5 by inserting the following elements (15)

3, 14, 7, 1, 8, 5, 11, 17, 13, 6, 23, 12, 20, 26, 4, 16, 18, 24, 25, and 19. Write the procedure for insertion and deletion. State the significance of B-tree.

KINGS COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
KEY FOR EVALUATION - MODEL EXAMINATION (OCT' 2019)

PART - A (10 x 2 = 20)

1. State the advantage of ADT

- ADT is reusable and ensures robust data structure.
- It reduces coding efforts.
- Encapsulation ensures that data cannot be corrupted.
- ADT is based on principles of Object Oriented Programming (OOP) and Software Engineering (SE).
- It specifies error conditions associated with operations.

2. Distinguish linear data structure with non-linear data structure.

Feature	Linear Data structure	Non-Linear Data structure
Basic	Data items are arranged in an orderly manner where the elements are attached adjacently (sequentially)	It arranges the data in a sorted order and there exists a relationship between the data elements
Traversing of the data	Data elements can be accessed in one time (single run)	Traversing of data elements in one go is not possible
Ease of implementation	Simpler	Complex
Levels involved	Single	Multiple
Example	Array, queue, stack, linked list	Tree and Graph

3. Tabulate the operations that can be performed on queue

Operations	Description
enqueue()	This function defines the operation for adding an element into queue
dequeue()	This function defines the operation for removing an element from queue
init()	This function is used for initializing the queue.
front	Front is used to get the front data item from a queue
rear	Rear is used to get the last item from a queue.

4. Convert $3*3/(4-1)+6*2$ expression into postfix form.

In a postfix expression, operators comes after operand. Hence,
 Postfix Expression is $33*41-/62*+$

5. Explain height of node & tree, depth of node, degree of node and edge of a tree.

Height of Node	Height of a node represents the number of edges on the longest path between that node and a leaf.
Height of Tree	Height of tree represents the height of its root node.
Depth of Node	Depth of a node represents the number of edges from the tree's root node to the node.
Degree of Node	Degree of a node represents a number of children of a node.
Edge	Edge is a connection between one node to another. It is a line between two nodes or a node and a leaf.

6. Enumerate on Binary tree properties.

- The maximum number of nodes at level 'l' = 2^{l-1} .
- Maximum number of nodes = $2^h - 1$. Here h is height of a tree. Height is considered as is maximum number of nodes on root to leaf path
- Minimum possible height = $\text{ceil}(\log_2(n+1))$

- In Binary tree, number of leaf nodes is always one more than nodes with two children.
 - Time Complexity of Tree Traversal: $O(n)$
- Write short note on Adjacency list graph representation
 - Adjacency list is a representation of graphs.
 - It is a collection of unordered list, used to represent a finite graphs.
 - Each list describes the set of neighbors of a vertex in the graph.
 - Adjacency list requires less amount of memory.
 - For every vertex, adjacency list stores a list of vertices, which are adjacent to the current one.
 - In adjacency list, an array of linked list is used. Size of the array is equal to the number of vertices.
 - When is an undirected graph called Biconnected?
 - An undirected graph is called Biconnected if there are two vertex-disjoint paths between any two vertices. In a Biconnected Graph, there is a simple cycle through any two vertices.
 - By convention, two nodes connected by an edge form a biconnected graph, but this does not verify the above properties. For a graph with more than two vertices, the above properties must be there for it to be Biconnected.
 - What is hashing?
 - Hashing is the process of mapping large amount of data item to smaller table with the help of hashing function.
 - Hashing is also known as **Hashing Algorithm** or **Message Digest Function**.
 - It is a technique to convert a range of key values into a range of indexes of an array
 - Give the advantage and disadvantage of separate chaining over linear probing.

Separate chaining

Pros

 - Simple to implement
 - Hash table capacity not limited
 - Robust technique.
 - Accommodate many keys

Cons

 - Cache performance is not good
 - Wastage of space
 - Search time can be $O(n)$ in worst case
 - Uses extra space for links

Linear probing

Pros

 - Best cache performance
 - Easy to compute

Cons

Suffers from clustering

PART - B (5 *13 = 65)

- State the polynomial representation for $6x^3+9x^2+7x+1$ using linked list. Write procedure to add and multiply two polynomial and explain with suitable example. (7)

Representation-2
Add -2
Multiplication-3
- What are the ways to insert a node in linked list? Write an algorithm for inserting a node before a given node in a linked list. (6)

5 options (first, last, prev.node, next.node, index loc)
(or)
- What are the various operations on array? Write a procedure to insert an element in the middle of the array. (7)

Insertion, Deletion, Search, Update

ii. Write a procedure to delete the last node from a circular linked list.
Delete last node – prenode → ptr = firstnode → preptr

(6)

12.a.i. Write procedure and explain linked list representation of Queue. Give procedure for conversion of Infix to Prefix expression.

(13)

Linked list representation-5

Infix to prefix expression – algorithm-4 & illustration-4
(or)

12.b. Describe the procedure for array and linked list implementation of Priority queue. Give the application of Queue & Dequeue.

(13)

Array representation-4

Linked list implementation-5

Application : 2+2

13.a.i. Write a routine for AVL tree insertion. Insert the following elements in the empty tree and how do you balance the tree after each element insertion? 2, 5, 4, 6, 7, 9, 8, 3, 1, 10. Brief about B+ tree. Discuss the application of heap.

(13)

Insertion algorithm-4

Balance – Rotation (Single, Double)-4

B+ tree-2

Heap application -2

(or)

13.b.i. Give algorithm and illustrate Inorder, Preorder and Postorder traversal of a tree. Write the routine for deletion of a node in Binary search tree. Give eg.

(13)

Inorder : L ROOT R

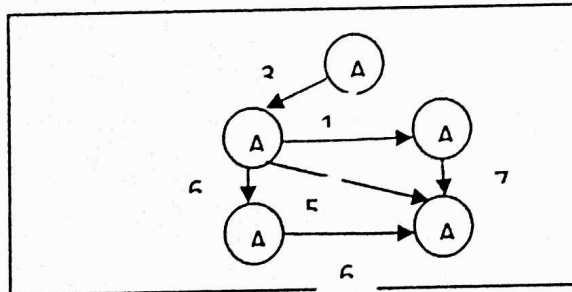
Preorder : LR ROOT

Postorder : ROOT LR

Binary Search tree – deletion – 3 cases (no child, single child, 2-child cases)

14.a.i. Apply an appropriate algorithm to find the shortest path from A to every other node of A, for the given graph below. Write the procedure.

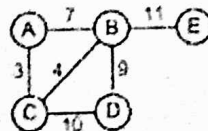
(7)



Dijkstra's algorithm

a.ii. Construct the MST of the graph given below and write the procedure.

(6)



MST procedure

MST graph

(or)

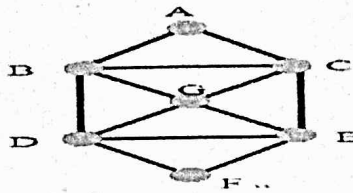
14.b.i. Illustrate with example Strongly Connected components.

(6)

A directed graph is strongly connected if there is a path between all pairs of vertices. A strongly connected component (SCC) of a directed graph is a maximal strongly connected subgraph eg.

ii. Find an Euler path and circuit using DFS for the following graph.

(7)



Euler path & circuit – visit all the nodes and return to the same node. splicing operation

15.a.i. Distinguish between linear search and binary search. State and explain the algorithms for both the search with example. (7)

Linear search -3

Binary search -4

ii. Sort the sequence of numbers 3,1,4,1,5,9,2,6,5 using Insertion sort. Explain the routine and its performance. (6)

INSERTION-SORT (ARR, N)

```

Step 1: Repeat Steps 2 to 5 for K = 1 to N - 1
Step 2:   SET TEMP = ARR[K]
Step 3:   SET J = K - 1
Step 4:   Repeat while TEMP <= ARR[J]
           SET ARR[J + 1] = ARR[J]
           SET J = J - 1
           [END OF INNER LOOP]
Step 5:   SET ARR[J + 1] = TEMP
           [END OF LOOP]
Step 6: EXIT
  
```

(or)

15.b.i. Sort the given integers and explain the intermediate results using shell sort.

35, 12, 14, 9, 15, 45, 32, 95, 40, 5. Explain the algorithm. Mention its complexity, & cons. (7)

pros

Shell_Sort(Arr, n)

```

Step 1: SET FLAG = 1, GAP_SIZE = N
Step 2: Repeat Steps 3 to 6 while FLAG = 1 OR GAP_SIZE > 1
Step 3:   SET FLAG = 0
Step 4:   SET GAP_SIZE = (GAP_SIZE + 1) / 2
Step 5:   Repeat Step 6 for I = 0 to I < (N - GAP_SIZE)
Step 6:   IF Arr[I + GAP_SIZE] > Arr[I]
           SWAP Arr[I + GAP_SIZE], Arr[I]
           SET FLAG = 1
Step 7: END
  
```

ii. Explain Rehashing and extendible hashing. (6)

Bucket, Directories – Dynamic hashing

Rehashing – doubling and apply indeed function

PART – C (1 * 15 = 15)

16.a. Write procedure to

Compute intersect of 2 lists

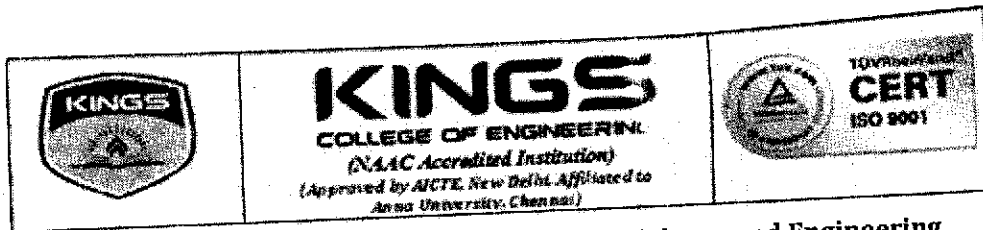
while(L1Pos != NULL && L2Pos != NULL)

```

{
  if( L1Pos->Element < L2Pos->Element )
    L1Pos = Next( L1Pos, L1 );
  else if( L1Pos->Element > L2Pos->Element )
    L2Pos = Next( L2Pos, L2 );
  else
  {
    Insert( L1Pos->Element, Result, ResultPos );
    L1 = Next( L1Pos, L1 ); L2 = Next( L2Pos, L2 );
    ResultPos = Next( ResultPos, Result );
  }
}
  
```

Reverse a list using non recursive procedure

(5)



Department of Department of Computer Science and Engineering
ACADEMIC YEAR 2019-2020/ Odd Semester
Weekly Test I - Mark Statement

YEAR / SEM / SEC : II - Year/ III SEM / A

STAFF : Abhirami.

COURSE : DATA STRUCTURES

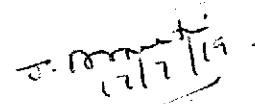
S.No	Register Number	Student Name	Marks	Retest Marks	Weekly Test I (20)
1	821118104001	ABARNA M	19		19
2	821118104002	ABINAYA N	20		20
3	821118104003	ABIRAMI K	19		19
4	821118104004	ABIRAMI M	17		17
5	821118104005	ABIRAMI P	20		20
6	821118104006	ABURVANAYAKI T	11	18	11
7	821118104007	AI SWARYA K	20		20
8	821118104008	ARCHANA A	19		19
9	821118104009	ARUNKUMAR G	16	20	16
10	821118104010	BABU M	6	19	6
11	821118104011	DEEPIKA V	00	19	0
12	821118104012	DHARSHINI T R	20		20
13	821118104013	DIVYABHARATHI G	20		20
14	821118104014	GAYATHRI J	16	20	16
15	821118104015	GAYATHRI K	19		19
16	821118104016	GUNASEELAN B	16	20	16
17	821118104017	HARIHARAN E	16	20	16
18	821118104018	ISHWARYA S	19		19
19	821118104019	JEMIMAESTHERGRACE M	10	18	10
20	821118104020	KAMALA PRIYA M	20		20
21	821118104021	KAVIYACEHLVAN K	8	17	8
22	821118104022	KEERTHANA V	18		18
23	821118104024	NANDHA KISHORE S	12	17	12
24	821118104025	NASRIN BANU N	20		20
25	821118104026	NATHIYADEVI N	10	20	10
26	821118104027	NAVEEN SUNDHAR V	6	17	6
27	821118104028	PATHMANABAN R	10	19	10
28	821118104029	PRIYA G	7	17	7
29	821118104030	PRIYADHARSHINI M	AB	17	0

S.No	Register Number	Student Name	Marks	Retest Marks	Weekly Test I (20)
30	821118104031	PRIYADHARSHINI R	20		20
31	821118104032	RAMYA J	11	20	11
32	821118104033	RATHNAKUMAR T I	15	18	15
33	821118104034	SARATHKUMAR R I	16	19	16
34	821118104035	SELVARANI T /	AB	19	0
35	821118104036	SIVARANJANI T A	OD	19	0 OD
36	821118104037	SNEHA K R /	12	17	12
37	821118104038	SNEKA S	20		20
38	821118104039	SOWMIYA K R	OD	18	0 OD
39	821118104040	SRINITHI C A	OD	19	0 OD
40	821118104041	SURUTHI S	9	20	9
41	821118104042	SURYA R A	9	20	9
42	821118104043	VAISHNAVI P A	OD	19	0 OD
43	821118104044	VITHYATHARAN S /	6	18	6
44	821119911982	GOPINATHAN D /	5	18	5
45	821119911983	SINDHU R * (Tocun) R	5	18	5

No. of students Present : 37 + 1
 No. of students Absent : 7 (AB + OD)
 No. of students Pass : 29
 No. of students Fail : 8 + 1
 Pass % : 65.91 (76.31 Excluding Abs)


 Staff Incharge 15/7/19


 HOD 15/7


 Principal

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Weekly Test I
CUMULATIVE MARK DETAILS - 2019-2020
Branch :Computer Science and Engineering
Year/Sem/Sec :II - Year - III SEM - A

			Date of Test	12/07/2019	08/07/2019	09/07/2019	10/07/2019	11/07/2019		
			Date of Submission	15/07/2019	11/07/2019	13/07/2019	13/07/2019	15/07/2019		
				EC8395-CE	MA8351-DM	CS8351-DPSD	CS8391-DS	CS8392-OOP		
SNO	Student Id	Student Name	Section	CE	DM	DPSD	DS	OOP	Total	Result
1	K18CS001	ABARNA M	A	18	16	13	19	17	83	PASS
2	K18CS002	ABINAYA N	A	17	18	17	20	10	82	PASS
3	K18CS003	ABIRAMI K	A	18	20	17	19	19	93	PASS
4	K18CS004	ABIRAMI M	A	19	16	13	17	16	81	PASS
5	K18CS005	ABIRAMI P	A	18	17	20	20	18	93	PASS
6	K18CS006	ABURVANAYAKI T	A	19	17	17	11	19	83	PASS
7	K18CS007	AISWARYA K	A	17	19	11	20	16	83	PASS
8	K18CS008	ARCHANA A	A	17	19	15	19	18	88	PASS
9	K18CS009	ARUNKUMAR G	A	13	AB	20	16	11	60	FAIL
10	K18CS010	BABU M	A	16	14	16	6	14	66	FAIL
11	K18CS011	DEEPIKA V	A	AB	OD	OD	OD	OD	0	FAIL
12	K18CS012	DHARSHINI T R	A	17	19	17	20	20	93	PASS
13	K18CS013	DIVYABHARATHI G	A	18	12	20	20	18	88	PASS
14	K18CS014	GAYATHRI J	A	17	20	20	16	19	92	PASS
15	K18CS015	GAYATHRI K	A	19	19	8	19	18	83	FAIL
16	K18CS016	GUNASEELAN B	A	12	AB	19	16	14	61	FAIL
17	K18CS017	HARIHARAN E	A	12	15	20	16	15	78	PASS
18	K18CS018	ISHWARYA S	A	18	18	16	19	17	88	PASS
19	K18CS019	JEMIMAESTHERGRACE M	A	AB	13	16	10	15	54	FAIL
20	K18CS020	KAMALA PRIYA M	A	18	16	17	20	19	90	PASS
21	K18CS021	KAVIYACEHLVAN K	A	13	15	16	8	15	67	FAIL
22	K18CS022	KEERTHANA V	A	18	17	19	18	15	87	PASS
23	K18CS024	NANDHA KISHORE S	A	5	10	10	12	12	49	FAIL
24	K18CS025	NASRIN BANU N	A	19	18	19	20	17	93	PASS
25	K18CS026	NATHIYADEVI N	A	17	13	18	10	15	73	PASS
26	K18CS027	NAVEEN SUNDHAR V	A	10	8	6	6	4	34	FAIL
27	K18CS028	PATHMANABAN R	A	13	19	19	10	16	77	PASS
28	K18CS029	PRIYA G	A	13	7	AB	7	11	38	FAIL
29	K18CS030	PRIYADHARSHINI M	A	14	17	AB	AB	16	47	FAIL
30	K18CS031	PRIYADHARSHINI R	A	18	16	17	20	17	88	PASS
31	K18CS032	RAMYA J	A	15	11	18	11	14	69	PASS
32	K18CS033	RATHINAKUMAR T	A	13	9	18	15	9	64	FAIL
33	K18CS034	SARATHKUMAR R	A	10	19	18	16	11	74	PASS
34	K18CS035	SELVARANI T	A	18	19	15	AB	20	72	FAIL
35	K18CS036	SIVARANJANI T	A	10	OD	OD	OD	OD	10	FAIL
36	K18CS037	SNEHA K	A	15	13	15	12	14	69	PASS
37	K18CS038	SNEKA S	A	19	19	13	20	16	87	PASS

38	K18CS039	SOWMIYA K	A	AB	OD	OD	OD	OD	0	FAIL
39	K18CS040	SRINITHI C	A	AB	OD	OD	OD	OD	0	FAIL
40	K18CS041	SURUTHI S	A	17	15	17	9	19	77	FAIL
41	K18CS042	SURYA R	A	10	6	4	9	12	41	FAIL
42	K18CS043	VAISHNAVI P	A	AB	OD	OD	OD	OD	0	FAIL
43	K18CS044	VITHYATHARAN S	A	10	AB	17	6	16	49	FAIL
44	KL19CS01	GOPINATHAN D	A	AB	AB	1	5	AB	6	FAIL
45	KT19CS01	SINDHU R	A	12		AB	5	14	31	FAIL

No.of Students Passed : 23

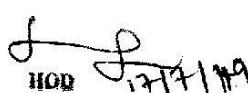
No.of Students Failed : 22

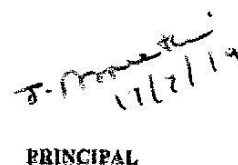
Over All Pass Percentage : 51.11

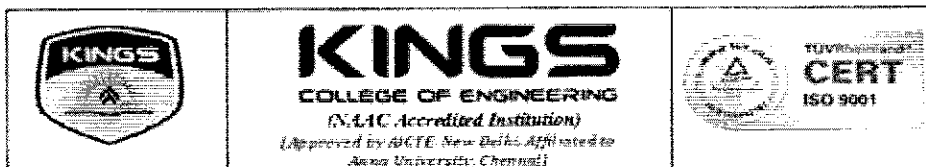
Desc	CE	DM	DPSD	DS	OOP
Total No.of Students	45	44	45	45	45
No.of Students Passed	38	31	33	29	37
No.of Students Failed	7	13	12	16	8
No.of Students Absent	6	9	8	7	6
No.of Students Secured < 16	23	23	20	24	24
Class Avg	13.16	12.25	12.71	12.04	13.24
Pass Percentage	97.44	88.57	89.19	76.32	94.87

Course Code	Course ShortCode	Course Name	Staff Name
EC8395	CE	COMMUNICATION ENGINEERING	T.Jeyaseelan
MA8351	DM	DISCRETE MATHEMATICS	Geetha S
CS8351	DPSD	DIGITAL PRINCIPLES AND SYSTEM DESIGN	Rajarajan.S
CS8391	DS	DATA STRUCTURES	Abhirami.K
CS8392	OOP	OBJECT ORIENTED PROGRAMMING	Sivakumar.D


STAFF


HOD 12/7/19


PRINCIPAL 11/7/19



Department of Computer Science and Engineering

ACADEMIC YEAR 2019-2020 (III SEM)

Assessment Test I EXAM - 24/07/2019

II - Year CSE 'A' [CS8391 - DATA STRUCTURES]

Conducted Period : 20

Roll No.	Name of the Student	Attn Period	Attn %	Attn Mark (10)	Assign Mark (20)	WT Mark Out of 20	AT Mark Out of 50	Tot Mark Out of 70
1	ABARNA M	20	100	10	20	19	31	51
2	ABINAYA N	20	100	10	20	20	30	50
3	ABIRAMI K	19	95	10	20	19	49	69
4	ABIRAMI M	18	90	9	20	17	37	57
5	ABIRAMI P	20	100	10	20	20	46	66
6	ABURVANAYAKI T	20	100	10	20	18	42	62
7	AISWARYA K	19	95	10	20	20	48	68
8	ARCHANA A	20	100	10	20	19	32	52
9	ARUNKUMAR G	15	75	8	20	20	38	58
10	BABU M	20	100	10	20	19	30	50
11	DEEPIKA V	19	95	10	20	19	39	59
12	DHARSHINI T R	20	100	10	20	20	35	55
13	DIVYABHARATHI G	20	100	10	20	20	42	62
14	GAYATHRI I	20	100	10	20	20	49	69
15	GAYATHRI K	20	100	10	20	19	41	61
16	GUNASEELAN B	17	85	9	20	20	44	64
17	HARIHARAN E	18	90	9	20	20	47	67
18	ISHWARYA S	19	95	10	20	19	46	66
19	JEMIMA ESTHER GRACE M	15	75	8	20	18	38	58
20	KAMALA PRIYA M	20	100	10	20	20	40	60
21	KAVIYACEHLVAN K	18	90	9	20	17	35	55
22	KEERTHANA V	20	100	10	20	18	36	56
23	NANDHA KISHORE S	15	75	8	20	17	38	58
24	NASRIN BANU N	17	85	9	20	20	35	55
25	NATHIYADEVI N	15	75	8	20	20	38	58

Roll No.	Name of the Student	Attn Period	Attn %	Attn Mark (10)	Assign Mark (20)	WT Mark Out of 20	AT Mark Out of 50	Tot Mark Out of 70
26	NAVEEN SUNDHAR V	17	85	9	20	17	35	55
27	PATHMANABAN R	20	100	10	20	19	36	56
28	PRIYA G	13	65	7	20	17	35	55
29	PRIYADHARSHINI M	14	70	7	20	17	37	57
30	PRIYADHARSHINI R	19	95	10	20	20	48	68
31	RAMYA I	20	100	10	20	20	35	55
32	RATHNAKUMAR T	20	100	10	20	18	32	52
33	SARATHKUMAR R	18	90	9	20	19	31	51
34	SELVARANI T	18	90	9	20	17	42	62
35	SIVARANJANI T	20	100	10	20	19	38	58
36	SNEHA K	17	85	9	20	17	34	54
37	SNEKA S	20	100	10	20	20	31	51
38	SOWMIYA K	19	95	10	20	18	36	56
39	SRINITHI C	17	85	9	20	19	36	56
40	SURUTHI S	19	95	10	20	20	30	50
41	SURYA R	14	70	7	20	20	35	55
42	VAISHNAVI P	19	95	10	20	19	30	50
43	VITHYATHARAN S	19	95	10	20	18	30	50
44	GOPINATHAN D	10	50	5	0	18		
45	SINDHU R	7	35	4	20	18	35	55

	Out of 50	Out of 70
Total	45	45
Absent	1	1
Appeared	44	44
Fail	0	1
Passed	44	44
Pass %	100	98
60 - 80	-	13
80 - 100	-	0
>= 90	-	0


Staff Incharge


HOD / CSE


Principal

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Assessment Test I
CUMULATIVE MARK DETAILS - 2019-2020
Branch :Computer Science and Engineering
Year/Sem/Sec :II - Year - III SEM - A

			Date of Test	26/07/2019	22/07/2019	23/07/2019	24/07/2019	25/07/2019		
			Date of Submission	29/07/2019	24/07/2019	26/07/2019	27/07/2019	29/07/2019		
				EC8395-CE	MA8351-DM	CS8351-DPSD	CS8391-DS	CS8392-OOP		
SNO	Student Id	Student Name	Section	CE	DM	DPSD	DS	OOP	Total	Result
1	K18CS001	ABARNA M	A	36	37	31	31	39	174	PASS
2	K18CS002	ABINAYA N	A	43	41	32	30	22	168	FAIL
3	K18CS003	ABIRAMI K	A	43	49	44	49	45	230	PASS
4	K18CS004	ABIRAMI M	A	31	30	24	37	31	153	FAIL
5	K18CS005	ABIRAMI P	A	43	47	38	46	40	214	PASS
6	K18CS006	ABURVANAYAKI T	A	45	48	37	42	36	208	PASS
7	K18CS007	AISWARYA K	A	40	30	15	48	38	171	FAIL
8	K18CS008	ARCHANA A	A	36	33	30	32	31	162	PASS
9	K18CS009	ARUNKUMAR G	A	41	16	30	38	30	155	FAIL
10	K18CS010	BABU M	A	25	18	37	30	17	127	FAIL
11	K18CS011	DEEPIKA V	A	30	33	20	22	31	136	FAIL
12	K18CS012	DHARSHINI T R	A	34	31	30	35	32	162	PASS
13	K18CS013	DIVYABHARATHI G	A	45	40	30	42	39	196	PASS
14	K18CS014	GAYATHRI J	A	37	47	38	49	49	220	PASS
15	K18CS015	GAYATHRI K	A	42	46	39	41	39	207	PASS
16	K18CS016	GUNASEELAN B	A	36	7	28	44	35	150	FAIL
17	K18CS017	HARIHARAN E	A	38	39	37	47	37	198	PASS
18	K18CS018	ISHWARYA S	A	44	45	34	46	40	209	PASS
19	K18CS019	JEMIMA ESTHER GRACE M	A	21	25	23	19	14	102	FAIL
20	K18CS020	KAMALA PRIYA M	A	34	41	31	27	37	170	FAIL
21	K18CS021	KAVIYACEHLVAN K	A	20	8	41	27	20	116	FAIL
22	K18CS022	KEERTHANA V	A	39	44	37	36	22	178	FAIL
23	K18CS024	NANDHA KISHORE S	A	28	25	21	18	24	116	FAIL
24	K18CS025	NASRIN BANU N	A	38	34	39	24	25	160	FAIL
25	K18CS026	NATHIYADEVI N	A	35	35	35	38	37	180	PASS
26	K18CS027	NAVEEN SUNDHAR V	A	16	5	32	7	6	66	FAIL
27	K18CS028	PATHMANABAN R	A	33	22	41	36	19	151	FAIL
28	K18CS029	PRIYA G	A	14	24	28	23	AB	89	FAIL
29	K18CS030	PRIYADHARSHINI M	A	24	22	14	24	18	102	FAIL
30	K18CS031	PRIYADHARSHINI R	A	43	43	43	48	31	208	PASS
31	K18CS032	RAMYA J	A	35	30	30	19	17	131	FAIL
32	K18CS033	RATHNAKUMAR T	A	32	18	41	32	26	149	FAIL
33	K18CS034	SARATHKUMAR R	A	30	18	38	31	30	147	FAIL
34	K18CS035	SELVARANI T	A	35	41	35	42	31	184	PASS
35	K18CS036	SIVARANJANI T	A	23	23	26	29	25	126	FAIL
36	K18CS037	SNEHA K	A	33	AB	17	18	19	87	FAIL
37	K18CS038	SNEKA S	A	42	23	30	31	36	162	FAIL

38	K18CS039	SOWMIYA K	A	25	22	20	25	19	111	FAIL
39	K18CS040	SRINITHI C	A	32	17	35	27	21	132	FAIL
40	K18CS041	SURUTHI S	A	32	30	36	30	31	159	PASS
41	K18CS042	SURYA R	A	29	32	3	35	26	125	FAIL
42	K18CS043	VAISHNAVI P	A	37	41	40	30	39	187	PASS
43	K18CS044	VITHYATHARAN S	A	34	13	36	30	32	145	FAIL
44	KL19CS01	GOPINATHAN D	A	AB	AB	AB	AB	AB	0	FAIL
45	KT19CS01	SINDHU R	A	22	17	25	15	13	92	FAIL

No. of Students Passed : 16

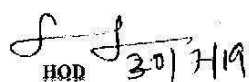
No. of Students Failed : 29

Over All Pass Percentage : 35.56

Desc	CE	DM	DPSD	DS	OOP
Total No. of Students	45	45	45	45	45
No. of Students Passed	33	25	31	29	25
No. of Students Failed	12	20	14	16	20
No. of Students Absent	1	2	1	1	2
No. of Students Secured (41-49)	10	12	5	12	2
No. of Students Secured (21-40)	31	21	33	26	31
No. of Students Secured < 16	2	6	4	3	5
Class Avg	32.78	28.67	30.47	31.78	27.76
Pass Percentage	75	58.14	70.45	65.91	58.14

Course Code	Course ShortCode	Course Name	Staff Name
EC8395	CE	COMMUNICATION ENGINEERING	T.Jeyaseelan
MA8351	DM	DISCRETE MATHEMATICS	Geetha S
CS8351	DPSD	DIGITAL PRINCIPLES AND SYSTEM DESIGN	Rajarajan.S
CS8391	DS	DATA STRUCTURES	Abhirami.K
CS8392	OOP	OBJECT ORIENTED PROGRAMMING	Sivakumar.D


STAFF


HOD 30/7/19


PRINCIPAL



Department of Department of Computer Science and Engineering
ACADEMIC YEAR 2019-2020/ Odd Semester
Weekly Test II - Mark Statement

YEAR / SEM / SEC : II - Year/ III SEM / A

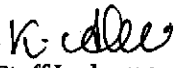
COURSE : DATA STRUCTURES

STAFF : Abhirami.K

S.No	Register Number	Student Name	Marks	Retest Marks	Weekly Test II (20)
1	821118104001	ABARNA M	20		20
2	821118104002	ABINAYA N	20		20
3	821118104003	ABIRAMI K	20		20
4	821118104004	ABIRAMI M	20		20
5	821118104005	ABIRAMI P	20		20
6	821118104006	ABURVANAYAKI T	20		20
7	821118104007	AISWARYA K	20		20
8	821118104008	ARCHANA A	20		20
9	821118104009	ARUNKUMAR G	20		20
10	821118104010	BABU M	19		19
11	821118104011	DEEPIKA V	20		20
12	821118104012	DHARSHINI T R	20		20
13	821118104013	DIVYABHARATHI G	20		20
14	821118104014	GAYATHRI J	20		20
15	821118104015	GAYATHRI K	20		20
16	821118104016	GUNASEELAN B	20		20
17	821118104017	HARIHARAN E	20		20
18	821118104018	ISHWARYA S	20		20
19	821118104019	JEMIMA ESTHER GRACE M	19		19
20	821118104020	KAMALA PRIYA M	20		20
21	821118104021	KAVIYACEHLVAN K	16		16
22	821118104022	KEERTHANA V	20		20
23	821118104024	NANDHA KISHORE S	AB		0
24	821118104025	NASRIN BANU N	20		20
25	821118104026	NATHIYADEVI N	20		20
26	821118104027	NAVEEN SUNDHAR V	16		16
27	821118104028	PATHMANABAN R	20		20
28	821118104029	PRIYA G	19		19
29	821118104030	PRIYADHARSHINI M	20		20

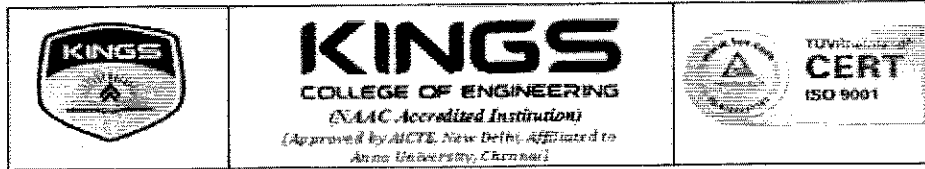
S.No	Register Number	Student Name	Marks	Retest Marks	Weekly Test II (20)
30	821118104031	PRIYADHARSHINI R	20		20
31	821118104032	RAMYA J	17		17
32	821118104033	RATHNAKUMAR T	20		20
33	821118104034	SARATHKUMAR R	19		19
34	821118104035	SELVARANI T	20		20
35	821118104036	SIVARANJANI T	20		20
36	821118104037	SNEHA K	17		17
37	821118104038	SNEKA S	7		7
38	821118104039	SOWMIYA K	17		17
39	821118104040	SRINITHI C	20		20
40	821118104041	SURUTHI S	AB		0
41	821118104042	SURYA R	20		20
42	821118104043	VAISHNAVI P	20		20
43	821118104044	VITHYATHARAN S	AB		0
44	821119911982	GOPINATHAN D	AB		0
45	821119911983	SINDHU R	14		14

No. of students Present : 41
 No. of students Absent : 4
 No. of students Pass : 40
 No. of students Fail : 1
 Pass % : 88.89


 Staff Incharge


 HOD


 Principal



Department of Department of Computer Science and Engineering
ACADEMIC YEAR 2019-2020/ Odd Semester
Weekly Test II - Mark Statement

YEAR / SEM / SEC : II - Year/ III SEM / A

COURSE : DATA STRUCTURES

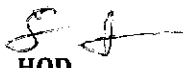
STAFF : Abhirami.K

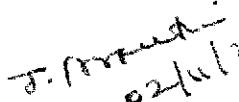
S.No	Register Number	Student Name	Marks	Retest Marks	Weekly Test II (20)
1	821118104001	ABARNA M	20		20
2	821118104002	ABINAYA N	20		20
3	821118104003	ABIRAMI K	20		20
4	821118104004	ABIRAMI M	20		20
5	821118104005	ABIRAMI P	20		20
6	821118104006	ABURVANAYAKI T	20		20
7	821118104007	AISWARYA K	20		20
8	821118104008	ARCHANA A	20		20
9	821118104009	ARUNKUMAR G	20		20
10	821118104010	BABU M	19		19
11	821118104011	DEEPIKA V	20		20
12	821118104012	DHARSHINI T R	20		20
13	821118104013	DIVYABHARATHI G	20		20
14	821118104014	GAYATHRI J	20		20
15	821118104015	GAYATHRI K	20		20
16	821118104016	GUNASEELAN B	20		20
17	821118104017	HARIHARAN E	20		20
18	821118104018	ISHWARYA S	20		20
19	821118104019	JEMIMA ESTHER GRACE M	19		19
20	821118104020	KAMALA PRIYA M	20		20
21	821118104021	KAVIYACEHLVAN K	16		16
22	821118104022	KEERTHANA V	20		20
23	821118104024	NANDHA KISHORE S	AB		0
24	821118104025	NASRIN BANU N	20		20
25	821118104026	NATHIYADEVI N	20		20
26	821118104027	NAVEEN SUNDHAR V	16		16
27	821118104028	PATHMANABAN R	20		20
28	821118104029	PRIYA G	19		19
29	821118104030	PRIYADHARSHINI M	20		20

S.No	Register Number	Student Name	Marks	Retest Marks	Weekly Test II (20)
30	821118104031	PRIYADHARSHINI R	20		20
31	821118104032	RAMYA J	17		17
32	821118104033	RATHNAKUMAR T	20		20
33	821118104034	SARATHKUMAR R	19		19
34	821118104035	SELVARANI T	20		20
35	821118104036	SIVARANJANI T	20		20
36	821118104037	SNEHA K	17		17
37	821118104038	SNEKA S	7		7
38	821118104039	SOWMIYA K	17		17
39	821118104040	SRINITHI C	20		20
40	821118104041	SURUTHI S	AB		0
41	821118104042	SURYA R	20		20
42	821118104043	VAISHNAVI P	20		20
43	821118104044	VITHYATHARAN S	AB		0
44	821119911982	GOPINATHAN D	AB		0
45	821119911983	SINDHU R	14		14

No. of students Present : 41
 No. of students Absent : 4
 No. of students Pass : 40
 No. of students Fail : 1
 Pass % : 88.89


 Staff Incharge


 HOD


 02/11/2019
 Principal

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Weekly Test II

CUMULATIVE MARK DETAILS - 2019-2020

Branch :Computer Science and Engineering

Year/Sem/Sec :II - Year - III SEM

			Date of Test	09/08/2019	05/08/2019	06/08/2019	07/08/2019	08/08/2019		
			Date of Submission	13/08/2019	07/08/2019	09/08/2019	13/08/2019	13/08/2019		
				EC8395-CE	MA8351-DM	CS8351-DPSD	CS8391-DS	CS8392-OOP		
SNO	Student Id	Student Name	Section	CE	DM	DPSD	DS	OOP	Total	Result
1	K18CS001	ABARNA M	A	19	18	18	20	18	91	PASS
2	K18CS002	ABINAYA N	A	18	18	18	20	17	88	PASS
3	K18CS003	ABIRAMI K	A	19	18	18	20	19	94	PASS
4	K18CS004	ABIRAMI M	A	18	17	18	20	17	79	PASS
5	K18CS005	ABIRAMI P	A	19	18	18	20	19	94	PASS
6	K18CS006	ABURVANAYAKI T	A	18	18	18	20	19	91	PASS
7	K18CS007	AISWARYA K	A	19	18	18	20	19	92	PASS
8	K18CS008	ARCHANA A	A	18	18	18	20	17	89	PASS
9	K18CS009	ARUNKUMAR G	A	19	18	17	20	19	93	PASS
10	K18CS010	BABU M	A	18	17	18	19	17	73	PASS
11	K18CS011	DEEPIKA V	A	19	18	18	20	17	90	PASS
12	K18CS012	DHARSHINI T R	A	18	18	18	20	18	92	PASS
13	K18CS013	DIVYABHARATHI G	A	18	18	18	20	19	93	PASS
14	K18CS014	GAYATHRI J	A	19	18	18	20	19	94	PASS
15	K18CS015	GAYATHRI K	A	19	18	18	20	17	90	PASS
16	K18CS016	GUNASEELAN B	A	19	18	17	20	19	93	PASS
17	K18CS017	HARIHARAN E	A	19	18	18	20	17	92	PASS
18	K18CS018	ISHWARYA S	A	19	18	17	20	18	92	PASS
19	K18CS019	JEMIMA ESTHER GRACE M	A	18	18	18	19	17	87	PASS
20	K18CS020	KAMALA PRIYA M	A	19	19	18	20	17	93	PASS
21	K18CS021	KAVIYACEHLVAN K	A	18	17	17	16	17	70	PASS
22	K18CS022	KEERTHANA V	A	19	18	18	20	18	91	PASS
23	K18CS024	NANDHA KISHORE S	A	17	17	18	AB	17	56	FAIL
24	K18CS025	NASRIN BANU N	A	19	18	18	20	17	89	PASS
25	K18CS026	NATHIYADEVI N	A	19	17	18	20	18	90	PASS
26	K18CS027	NAVEEN SUNDHAR V	A	17	17	18	16	17	53	PASS
27	K18CS028	PATHMANABAN R	A	18	17	18	20	17	82	PASS
28	K18CS029	PRIYA G	A	17	18	18	19	17	84	PASS
29	K18CS030	PRIYADHARSHINI M	A	18	18	17	20	17	86	PASS
30	K18CS031	PRIYADHARSHINI R	A	19	18	17	20	17	91	PASS
31	K18CS032	RAMYA J	A	19	18	18	17	19	75	PASS
32	K18CS033	RATHNAKUMAR T	A	18	18	18	20	19	84	PASS
33	K18CS034	SARATHKUMAR R	A	18	17	18	19	18	65	PASS
34	K18CS035	SELVARANI T	A	19	18	18	20	19	92	PASS
35	K18CS036	SIVARANJANI T	A	19	18	18	20	17	89	PASS
36	K18CS037	SNEHA K	A	19	17	18	17	18	62	PASS
37	K18CS038	SNFKA S	A	19	18	17	7	19	76	FAIL

38	K18CS039	SOWMIYA K	A	18	19	18	17	19	82	PASS
39	K18CS040	SRINITHI C	A	19	18	18	20	17	83	PASS
40	K18CS041	SURUTHI S	A	19	18	18	AB	18	37	FAIL
41	K18CS042	SURYA R	A	18	17	18	20	17	77	PASS
42	K18CS043	VAISHNAVI P	A	19	18	18	20	17	91	PASS
43	K18CS044	VITHYATHARAN S	A	17	17	18	AB	17	15	FAIL
44	KL19CS01	GOPINATHAN D	A	17	17	18	AB	17	0	FAIL
45	KT19CS01	SINDHU R	A	18	18	17	14	17	80	PASS

No. of Students Passed : 40 No. of Students Failed : 5 Over All Pass Percentage : 88.89

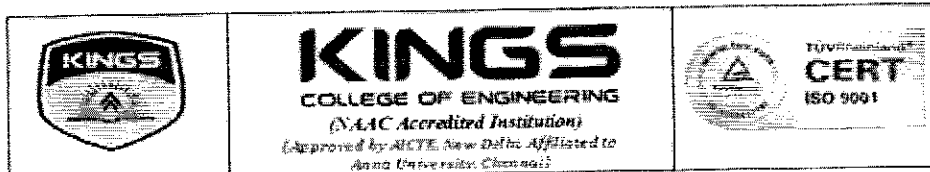
Desc	CE	DM	DPSD	DS	OOP
Total No. of Students	45	45	45	45	45
No. of Students Passed	45	45	45	40	45
No. of Students Failed	0	0	0	1	0
No. of Students Absent	0	0	0	4	0
No. of Students Secured < 16	0	0	0	6	0
Class Avg	18.42	17.78	17.82	17.33	17.76
Pass Percentage	100	100	100	97.56	100

Course Code	Course_ShortCode	Course Name	Staff Name	Section
EC8395	CE	COMMUNICATION ENGINEERING	T.Jeyaseelan	A
MA8351	DM	DISCRETE MATHEMATICS	Geetha S	A
CS8351	DPSD	DIGITAL PRINCIPLES AND SYSTEM DESIGN	Rajarajan.S	A
CS8391	DS	DATA STRUCTURES	Abhirami.K	A
CS8392	OOP	OBJECT ORIENTED PROGRAMMING	Sivakumar.D	A


STAFF


HOD 19/8/2019


19/8/2019
PRINCIPAL



Department of Computer Science and Engineering

ACADEMIC YEAR 2019-2020 (III SEM)

Assessment Test II EXAM - 06/09/2019

II - Year CSE 'A' [CS8391 - DATA STRUCTURES]

Conducted Period : 26

Roll No.	Name of the Student	Attn Period	Attn %	Attn Mark (10)	Assign Mark (20)	WT Mark Out of 20	AT Mark Out of 80	Tot Mark Out of 100
1	ABARNA M	26	100	10	20	20	56	76
2	ABINAYA N	25	96	10	20	20	40	60
3	ABIRAMI K	26	100	10	20	20	44	64
4	ABIRAMI M	26	100	10	20	20		56
5	ABIRAMI P	26	100	10	20	20	50	70
6	ABURVANAYAKI T	25	96	10	20	20	44	64
7	AISWARYA K	25	96	10	20	20	56	76
8	ARCHANA A	26	100	10	20	20	48	68
9	ARUNKUMAR G	21	81	8	20	20	40	60
10	BABU M	25	96	10	20	19		52
11	DEEPIKA V	26	100	10	20	20		54
12	DHARSHINI T R	25	96	10	20	20	40	60
13	DIVYABHARATHI G	26	100	10	20	20	52	72
14	GAYATHRI I	26	100	10	20	20	44	64
15	GAYATHRI K	26	100	10	20	20	40	60
16	GUNASEELAN B	24	92	9	20	20	48	68
17	HARIHARAN E	23	88	9	20	20	44	64
18	ISHWARYA S	23	88	9	20	20	46	66
19	JEMIMA ESTHER GRACE M	23	88	9	20	19		
20	KAMALA PRIYA M	26	100	10	20	20	44	64
21	KAVIYACEHLVAN K	26	100	10	20	16		55
22	KEERTHANA V	26	100	10	20	20	49	69
23	NANDHA KISHORE S	23	88	9	20			
24	NASRIN BANU N	26	100	10	20	20	40	60
25	NATHIYADEVI N	25	96	10	20	20	54	74

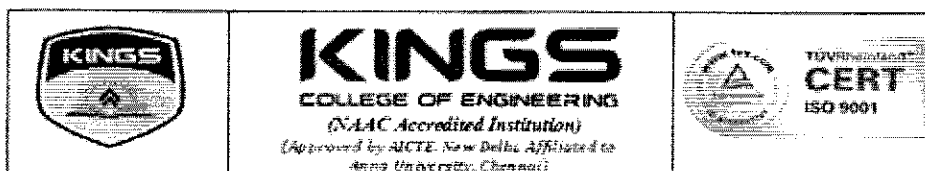
Roll No.	Name of the Student	Attn Period	Attn %	Attn Mark (10)	Assign Mark (20)	WT Mark Out of 20	AT Mark Out of 80	Tot Mark Out of 100
26	NAVEEN SUNDHAR V	23	88	9	20	16		
27	PATHMANABAN R	24	92	9	20	20	43	63
28	PRIYA G	26	100	10	20	19	40	60
29	PRIYADHARSHINI M	19	73	7	20	20		
30	PRIYADHARSHINI R	25	96	10	20	20	52	72
31	RAMYA J	26	100	10	20	17		
32	RATHNAKUMAR T	23	88	9	20	20	40	60
33	SARATHKUMAR R	23	88	9	20	19		55
34	SELVARANI T	21	81	8	20	20	56	76
35	SIVARANJANI T	26	100	10	20	20	50	70
36	SNEHA K	19	73	7	20	17		
37	SNEKA S	26	100	10	20			55
38	SOWMIYA K	26	100	10	20	17		55
39	SRINITHI C	24	92	9	20	20		57
40	SURUTHI S	20	77	8	20		52	72
41	SURYA R	21	81	8	20	20		54
42	VAISHNAVI P	26	100	10	20	20	40	60
43	VITHYATHARAN S	23	88	9	20			50
44	GOPINATHAN D	12	46	5	20			
45	SINDHU R	26	100	10	20	14		

	Out of 80	Out of 100
Total	45	45
Absent	0	0
Appeared	45	45
Fail	18	8
Passed	27	37
Pass %	60	82
60 - 80	-	27
80 - 100	-	0
>= 90	-	0

K. S. S. S.
Staff Incharge

[Signature]
HOD / CSE

[Signature]
02/11/2019
Principal



Department of Department of Computer Science and Engineering

ACADEMIC YEAR 2019-2020/ Odd Semester

MODEL EXAM - Mark Statement

YEAR / SEM / SEC : II - Year/ III SEM / A

COURSE : DATA STRUCTURES

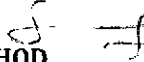
STAFF : Abhirami.K

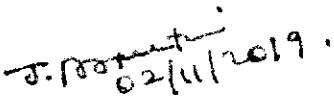
S.No	Register Number	Student Name	Marks	Retest Marks	MODEL EXAM (100)
1	821118104001	ABARNA M	52		52
2	821118104002	ABINAYA N	41		41
3	821118104003	ABIRAMI K	78		78
4	821118104004	ABIRAMI M	54		54
5	821118104005	ABIRAMI P	83		83
6	821118104006	ABURVANAYAKI T	72		72
7	821118104007	AISWARYA K	65		65
8	821118104008	ARCHANA A	41		41
9	821118104009	ARUNKUMAR G	63		63
10	821118104010	BABU M	44		44
11	821118104011	DEEPIKA V	39		39
12	821118104012	DHARSHINI T R	50		50
13	821118104013	DIVYABHARATHI G	75		75
14	821118104014	GAYATHRI J	66		66
15	821118104015	GAYATHRI K	62		62
16	821118104016	GUNASEELAN B	50		50
17	821118104017	HARIHARAN E	63		63
18	821118104018	ISHWARYA S	46		46
19	821118104019	JEMIMA ESTHER GRACE M	27		27
20	821118104020	KAMALA PRIYA M	22		22
21	821118104021	KAVIYACEHLVAN K	14		14
22	821118104022	KEERTHANA V	51		51
23	821118104024	NANDHA KISHORE S	11		11
24	821118104025	NASRIN BANU N	AB		0
25	821118104026	NATHIYADEVI N	39		39
26	821118104027	NAVEEN SUNDHAR V	01		1
27	821118104028	PATHMANABAN R	41		41
28	821118104029	PRIYA G	AB		0
29	821118104030	PRIYADHARSHINI M	32		32

S.No	Register Number	Student Name	Marks	Retest Marks	MODEL EXAM (100)
30	821118104031	PRIYADHARSHINI R	53		53
31	821118104032	RAMYA J	21		21
32	821118104033	RATHNAKUMAR T	33		33
33	821118104034	SARATHKUMAR R	32		32
34	821118104035	SELVARANI T	63		63
35	821118104036	SIVARANJANI T	36		36
36	821118104037	SNEHA K	19		19
37	821118104038	SNEKA S	44		44
38	821118104039	SOWMIYA K	35		35
39	821118104040	SRINITHI C	33		33
40	821118104041	SURUTHI S	43		43
41	821118104042	SURYA R	29		29
42	821118104043	VAISHNAVI P	23		23
43	821118104044	VITHYATHARAN S	18		18
44	821119911982	GOPINATHAN D	AB		0
45	821119911983	SINDHU R	12		12

No. of students Present : 42
 No. of students Absent : 3
 No. of students Pass : 16
 No. of students Fail : 26
 Pass % : 35.56


 Staff Incharge


 HOD


 02/11/2019
 Principal

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MODEL EXAM

CUMULATIVE MARK DETAILS - 2019-2020

Branch :Computer Science and Engineering

Year/Sem/Sec :II - Year - III SEM

			Date of Test	15/10/2019	10/10/2019	11/10/2019	12/10/2019	14/10/2019		
			Date of Submission	16/10/2019	11/10/2019	12/10/2019	14/10/2019	15/10/2019		
				EC8395-CE	MA8351-DM	CS8351-DPSD	CS8391-DS	CS8392-OOP		
SNO	Student Id	Student Name	Section	CE	DM	DPSD	DS	OOP	Total	Result
1	K18CS001	ABARNA M	A	55	63	42	52	78	290	FAIL
2	K18CS002	ABINAYA N	A	52	72	36	41	79	280	FAIL
3	K18CS003	ABIRAMI K	A	80	93	70	78	91	412	PASS
4	K18CS004	ABIRAMI M	A	50	69	61	54	84	318	PASS
5	K18CS005	ABIRAMI P	A	80	93	63	83	85	404	PASS
6	K18CS006	ABURVANAYAKI T	A	60	70	68	72	82	352	PASS
7	K18CS007	AISWARYA K	A	52	78	85	65	79	359	PASS
8	K18CS008	ARCHANA A	A	50	41	42	41	54	228	FAIL
9	K18CS009	ARUNKUMAR G	A	55	70	50	63	50	288	PASS
10	K18CS010	BABU M	A	AB	50	51	44	58	203	FAIL
11	K18CS011	DEEPIKA V	A	53	63	34	39	59	248	FAIL
12	K18CS012	DHARSHINI T R	A	51	70	53	50	72	296	PASS
13	K18CS013	DIVYABHARATHI G	A	75	89	84	75	84	407	PASS
14	K18CS014	GAYATHRI J	A	70	73	67	66	78	354	PASS
15	K18CS015	GAYATHRI K	A	60	82	61	62	82	347	PASS
16	K18CS016	GUNASEELAN B	A	36	66	65	50	69	286	FAIL
17	K18CS017	HARIHARAN E	A	37	59	64	63	53	276	FAIL
18	K18CS018	ISHWARYA S	A	56	61	71	46	67	301	FAIL
19	K18CS019	JEMIMA ESTHER GRACE M	A	32	51	57	27	32	199	FAIL
20	K18CS020	KAMALA PRIYA M	A	61	77	57	22	57	274	FAIL
21	K18CS021	KAVIYACEHLVAN K	A	36	29	43	14	20	142	FAIL
22	K18CS022	KEERTHANA V	A	72	78	72	51	69	342	PASS
23	K18CS024	NANDHA KISHORE S	A	16	27	30	11	19	103	FAIL
24	K18CS025	NASRIN BANU N	A	50	35	AB	AB	31	116	FAIL
25	K18CS026	NATHIYADEVI N	A	65	67	66	39	69	306	FAIL
26	K18CS027	NAVEEN SUNDHAR V	A	10	37	18	1	8	74	FAIL
27	K18CS028	PATHMANABAN R	A	50	64	65	41	50	270	FAIL
28	K18CS029	PRIYA G	A	32	AB	AB	AB	AB	32	FAIL
29	K18CS030	PRIYADHARSHINI M	A	30	55	62	32	56	235	FAIL
30	K18CS031	PRIYADHARSHINI R	A	50	63	70	53	81	317	PASS
31	K18CS032	RAMYA J	A	50	58	41	21	44	214	FAIL
32	K18CS033	RATHNAKUMAR T	A	40	52	39	33	50	214	FAIL
33	K18CS034	SARATHKUMAR R	A	42	65	43	32	39	221	FAIL
34	K18CS035	SELVARANI T	A	52	81	72	63	73	341	PASS
35	K18CS036	SIVARANJANI T	A	50	62	53	36	51	252	FAIL
36	K18CS037	SNEHA K	A	33	45	40	19	30	167	FAIL
37	K18CS038	SNEHA S	A	65	51	38	44	50	248	FAIL

38	K18CS039	SOWMIYA K	A	50	50	52	35	19	206	FAIL
39	K18CS040	SRINITHI C	A	25	67	50	33	21	196	FAIL
40	K18CS041	SURUTHI S	A	50	63	65	43	47	268	FAIL
41	K18CS042	SURYA R	A	AB	52	28	29	64	173	FAIL
42	K18CS043	VAISHNAVI P	A	50	75	43	23	33	224	FAIL
43	K18CS044	VITHYATHARAN S	A	21	55	56	18	47	197	FAIL
44	KL19CS01	GOPINATHAN D	A	5	7	10	AB	5	27	FAIL
45	KT19CS01	SINDHU R	A	40	56	41	12	27	176	FAIL

No.of Students Passed : 13 No.of Students Failed : 32 Over All Pass Percentage : 28.89

Desc	CE	DM	DPSD	DS	OOP
Total No.of Students	45	45	45	45	45
No.of Students Passed	28	37	27	16	29
No.of Students Failed	15	7	16	26	15
No.of Students Absent	2	1	2	3	1
No.of Students Secured (91-100)	0	2	0	0	1
No.of Students Secured (81-90)	0	3	2	1	6
No.of Students Secured (71-80)	4	6	3	3	6
No.of Students Secured (61-70)	4	15	13	6	5
No.of Students Secured (50-60)	20	11	9	6	11
No.of Students Secured < 50	17	8	18	29	16
Class Avg	45.53	59.64	50.62	39.47	53.24
Pass Percentage	65.12	84.09	62.79	38.10	65.91

Course Code	Course_ShortCode	Course Name	Staff Name	Section
EC8395	CE	COMMUNICATION ENGINEERING	T.Jeyaseelan	A
MA8351	DM	DISCRETE MATHEMATICS	Geetha S	A
CS8351	DPSD	DIGITAL PRINCIPLES AND SYSTEM DESIGN	Rajarajan.S	A
CS8391	DS	DATA STRUCTURES	Abhirami.K	A
CS8392	OOP	OBJECT ORIENTED PROGRAMMING	Sivakumar.D	A

STAFF

HOD

PRINCIPAL

KINGS COLLEGE OF ENGINEERING, PUNALKULAM

Branch : II CSE

Year / Semester : II / III

Date: 17.09.19

Consolidated Mark Statement: Week Test 1 / Assessment Test 1 / Week Test 2 / Assessment Test 2 / Model Exam

Roll No.	MA8351 4.09.19 DM		CS8351 05.09.19 DPSD		CS8391 06.09.19 DS		CS8392 07.09.19 OOP		EC8395 09.09.19 CE		Result
	80	100	80	100	80	100	80	100	80	100	
1.	40	50	28	36	56	71	64	80	48	61	FAIL
2.	45	57	30	38	40	50	48	61	43	54	FAIL
3.	67	84	68	86	44	55	65	82	65	82	PASS
4.	53	67	28	35	36	45	53	67	49	62	FAIL
5.	64	80	50	63	50	63	56	71	64	80	PASS
6.	64	81	44	55	44	56	65	82	60	75	PASS
7.	56	70	44	56	56	70	64	80	64	80	PASS
8.	45	57	40	50	48	60	48	61	52	65	PASS
9.	44	56	40	50	40	50	59	74	52	65	PASS
10.	46	58	29	37	32	40	36	45	41	52	FAIL
11.	45	57	40	50	34	43	47	59	56	70	FAIL
12.	53	67	26	33	40	50	54	68	40	50	FAIL
13.	63	79	57	72	52	65	69	87	72	90	PASS
14.	64	81	50	63	44	55	71	89	64	80	PASS
15.	56	71	59	74	40	50	59	74	52	65	PASS
16.	63	79	48	61	48	60	63	79	40	50	PASS
17.	63	79	32	40	44	56	52	66	32	41	FAIL
18.	58	73	48	61	46	58	66	83	60	75	PASS
19.	46	58	40	50	26	33	36	46	40	50	FAIL
20.	64	80	48	60	44	55	60	76	56	70	PASS
21.	0	01	15	19	35	44	25	32	32	41	FAIL
22.	61	77	57	72	49	62	53	67	52	66	PASS
23.	44	56	32	40	26	33	18	23	20	26	FAIL
24.	49	62	41	52	40	50	48	60	52	65	PASS
25.	61	77	46	58	54	68	66	83	52	65	PASS
26.	12	16	8	11	16	20	7	09	16	20	FAIL
27.	61	77	41	52	43	54	46	58	51	64	PASS
28.	44	56	40	50	40	50	32	41	40	50	FAIL
29.	37	47	25	32	28	35	53	67	40	50	FAIL
30.	60	76	43	54	52	66	69	87	40	51	PASS
31.	8	10	23	29	22	28	28	36	40	50	FAIL
32.	54	68	33	42	40	51	50	63	36	45	FAIL
33.	60	76	40	50	35	44	43	54	44	56	FAIL
34.	56	70	52	66	56	70	69	87	51	64	PASS
35.	48	61	40	50	50	63	56	71	56	70	PASS
36.	34	43	20	25	28	36	47	59	40	50	FAIL
37.	32	41	36	45	35	44	61	77	52	65	PASS
38.	43	54	32	40	35	44	31	39	40	50	FAIL

Roll No.	MA8351 4.09.19 DM		CS8351 05.09.19 DPSD		CS8391 06.09.19 DS		CS8392 07.09.19 OOP		EC8395 09.09.19 CE		Result
	80	100	80	100	80	100	80	100	80	100	
39.	43	54	41	52	37	47	60	76	36	45	FAIL
40.	45	57	47	59	52	65	62	78	41	52	PASS
41.	36	45	40	51	34	43	55	69	24	31	FAIL
42.	OD	OD	29	37	40	51	52	66	48	60	FAIL
43.	49	62	41	52	30	38	62	78	AB	AB	FAIL
44.	7	09	1	02	7	09	10	13	4	05	FAIL
45.	47	59	26	33	18	23	25	32	36	45	FAIL

Total No. of Students:45

Details	MA8351 4.09.19 DM	CS8351 05.09.19 DPSD	CS8391 06.09.19 DS	CS8392 07.09.19 OOP	EC8395 09.09.19 CE
Absent	1	0	0	0	1
Appeared	44	45	45	45	44
Pass	36	27	27	35	35
Fail	8	18	18	10	9
Pass %	81.82	60	60	77.78	79.55

SUB CODE	COURSE_SHORTCODE	SUBJET NAME	STAFF NAME
EC8395	CE	COMMUNICATION ENGINEERING	T.Jeyaseelan
MA8351	DM	DISCRETE MATHEMATICS	Geetha S
CS8351	DPSD	DIGITAL PRINCIPLES AND SYSTEM DESIGN	Rajarajan.S
CS8391	DS	DATA STRUCTURES	Abhirami.K
CS8392	OOP	OBJECT ORIENTED PROGRAMMING	Sivakumar.D

OVERALL PASS %=21/45=46.4

Signature of the coordinator

Name: (S.RAJARAJANI)

HOD / CSE

13/9/2019



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Course File

Format A - ASSIGNMENT

TITLE : Array Implementation of Stack, Stack Application

- Arithmetic Expression Conversion
- Polynomial Manipulation

OBJECTIVE : To learn about implementing stack operations
To gain knowledge on various stack applications

EVALUATION : 20 Marks

Part - A

- a) Linked list Implementation
- b) Stack application - Illustration
- c) Polynomial Manipulation

DATE OF COMPLETION : 17.7.2019

K. S. S. S. 7/8/19
Staff In-charge

A NAAC Accredited Institution



ASSIGNMENT CUM TEST NOTE

NAME : N. ABINAYA

REGISTER NO : 881118104002 ROLL NO : 02

DEPARTMENT : CSE

YEAR : II SEM : IV SEC :

SUBJECT : Data structures SUB CODE : CS8391

INDEX

Name of the Student : N. Abinaya

Register No : 821118104002 Year / Sec: II

Name of the Subject : Data Structures Sub Code: CS8391

ASSIGNMENT

S.No.	Topics	Date of Submission	Marks	Staff Sign	HOD/CC Signature
1	1) application of linked list 2) array and linked list implementation 3) stack.	17.7.19	20	Kedde	Kedde

TEST

Test	Date	Mark (Scored 100)	Staff Sign.	HOD/CC Signature
WT 1				
WT 2	7.8.19	20	Kedde	Kedde

Cross Test - III 21.8.19 20 Kedde Kedde

Remarks:

17.7.19

Page No. :

ASSIGNMENT - I (20 marks)

Before AT-I

Part - A:

① List down any four applications of data structures.

* Achieving data maintaining specific data structure helps improve data access and value.

Data structure and help program implement various programming.

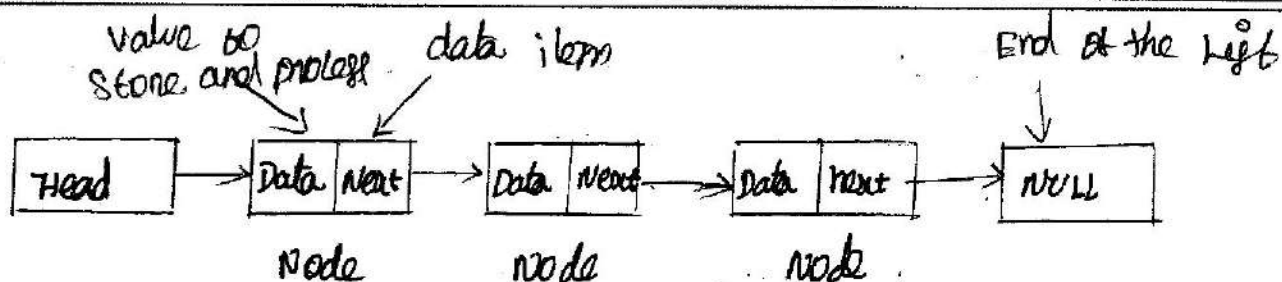
* Compiler design.

* Operating system.

② Define linked list:

* Linked list consists of series of nodes. Each node contains the element and a pointer to its successor node. The pointer of the last node points to null.

* Insertion and deletion operations are easily performed using linked list.



- ③ Given few example for data structures :
- Array, Stack, Tables
 - Linked list, tree
 - Decision tree, Graph and Forest.

- ④ What are the three ways to represent arithmetic expression?

Infix notation - $A / B + C$

postfix notation - $AB / C +$

prefix notation - $+ A / ABC$

- ⑤ Explain the usage of Stack in recursive algorithm implementation?

overflow :

Attempt to insert an element when the stack is full is said to be overflow.

underflow :

Attempt to delete an element when the stack is empty is said to be underflow.

Inserting node at the Beginning of a linked List :

STEP 1 : If $AVAIL = NULL$
 Write overflow
 GO TO STEP 7
 [END OF IF]

STEP 2 : SET $NEW_NODE = AVAIL$

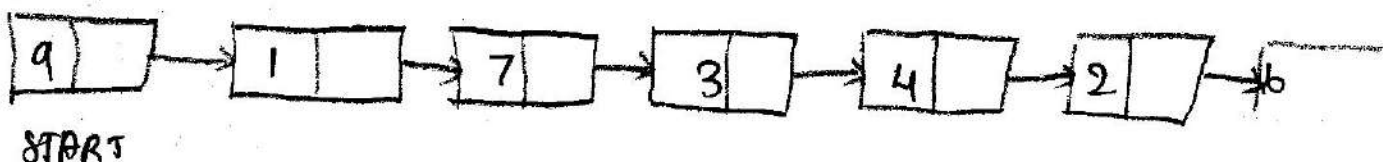
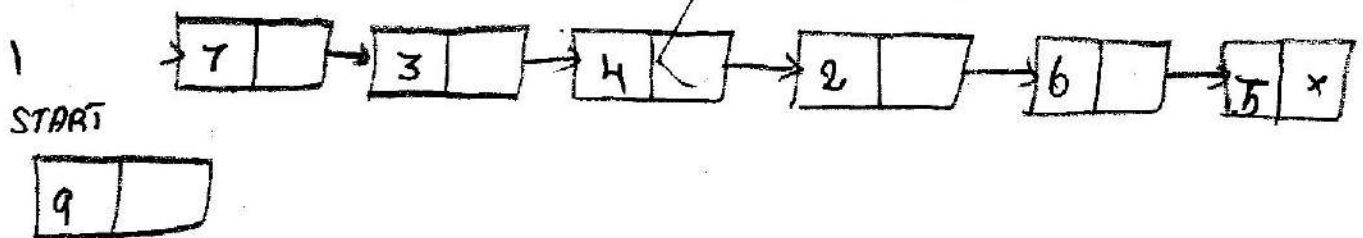
STEP 3 : SET $AVAIL = AVAIL \rightarrow NEXT$

STEP 4 : SET $NEW_NODE \rightarrow DATA = VAL$

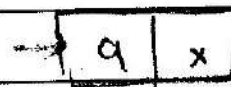
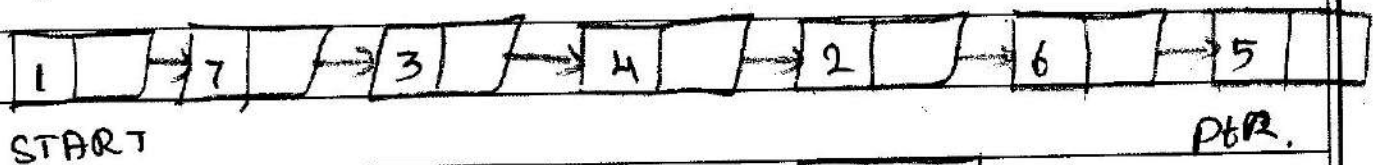
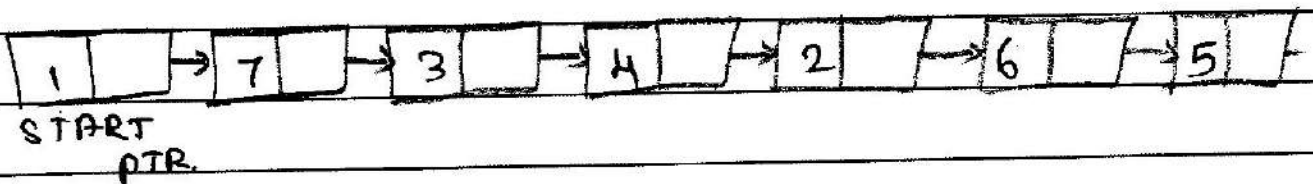
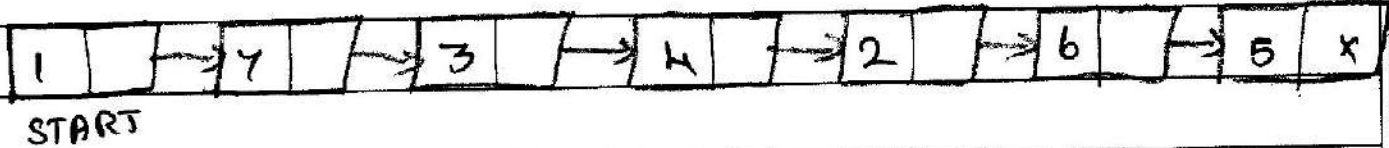
STEP 5 : SET $NEW_NODE \rightarrow NEXT = START$

STEP 6 : SET $START = NEW_NODE$

STEP 7 : Exit.



Inserting a Node at the END of a linked List.



STEP 1: IF $AVAIL = NULL$
 write OVERFLOW
 GO TO STEP 10
 [END OF IF]

20/20 Cool
 Real One

STEP 2: SET $NEW_NODE = AVAIL$

STEP 3: SET $AVAIL = AVAIL \rightarrow NEXT$

STEP 4: SET $NEW_NODE \rightarrow DATA = VAL$

STEP 5: SET $NEW_NODE \rightarrow NEXT = NULL$

STEP 6: SET $PTR = START$

STEP 7: Repeat STEP 8 while $PTR \rightarrow NEXT \neq NULL$

STEP 8: SET $PTR = PTR \rightarrow NEXT$

STEP 9: [END OF LOOP]

STEP 10: SET $PTR \rightarrow NEXT = NEW_NODE$

STEP 11: Both:

A NAAC Accredited Institution



ASSIGNMENT CUM TEST NOTE

NAME : Gr. Divyabharathi

REGISTER NO : 821118104013 ROLL NO : 13

DEPARTMENT : Computer Science & Engineering

YEAR : II SEM : III SEC :

SUBJECT : Data structure SUB CODE : CS8391

INDEX

Name of the Student : G. Divyabharathi

Register No : 821118104013 Year / Sec: II

Name of the Subject : Data Structure Sub Code: CS8391

ASSIGNMENT

S.No.	Topics	Date of Submission	Marks	Staff Sign	HOD/ce Signature
1.	Array implementation of stack application of stacks Infix to postfix, postfix to infix Polynomial Manipulation	17.7.2019	20	K. Reddy	K. Reddy

TEST

Test	Date	Mark (Scored 100) 20	Staff Sign.	HOD/ce Signature
WT 1				
WT 2	7.8.19	20	K. Reddy	K. Reddy

Remarks:

17.7.2019

ASSIGNMENT - 1

PART - A

1. List down any four applications of data structures:

Four applications of data structure:

Achieving and maintaining specific data structure
keep improve data access and value.

Data structure also help programmer implement
various programming tasks.

Compiler design

Operating system.

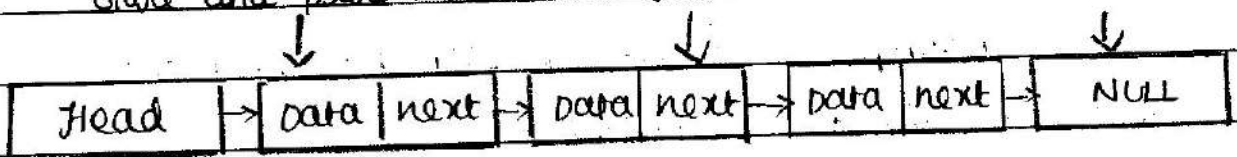
2. Define linked list:

Linked list

Linked list consists of series of nodes.

Each node contains the element and a pointer to its successor
node. The pointer of the last node points to NULL.

Actual value to	Address of the	End of the
store and process	next data item	list



node node node

PART-B

1. Explain the array and linked list implementation of stack.
Illustrate the application of stack.

Array implementation:

To push an element x into the stack, Top pointer is incremented and then set $\text{Stack}[\text{TOP}] = x$

To pop an element, the stack $[\text{TOP}]$ value is incremented and the top pointer is decremented.

pop an empty stack or push on a full stack will exceed the array bounds.

Push :

Step 1 : if $\text{top} = \text{max} - 1$
 write 'over-flow'
 go to step 4
 end if

Step 2 : set $\text{top} = \text{top} + 1$

Step 3 : set $A[\text{top}] = \text{val}$

Step 4 : End

Top operation returns the element at the front of list.

6: set $ptr = start$

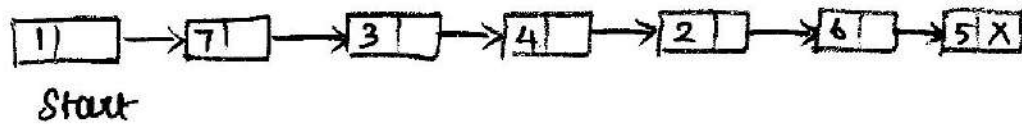
7: Repeat steps 8 while $ptr \rightarrow next \neq null$.

8: set $ptr \Rightarrow ptr \rightarrow next$

End of loop

9: $ptr \rightarrow next = new_node$

10: exit.

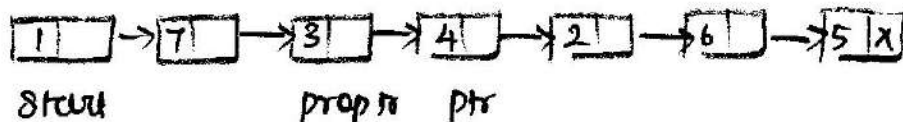
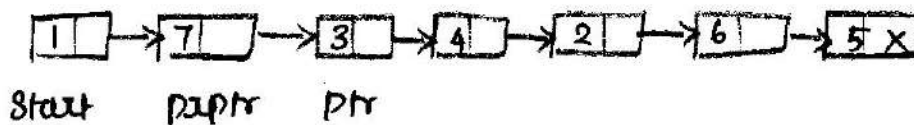
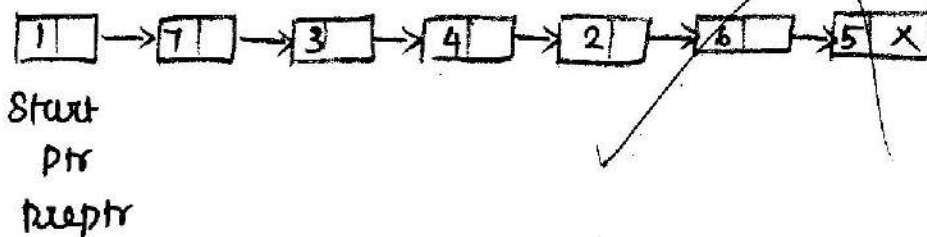


Allocate memory for the new_node and initialize its data to 9

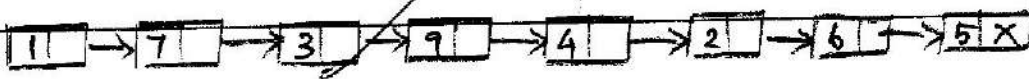
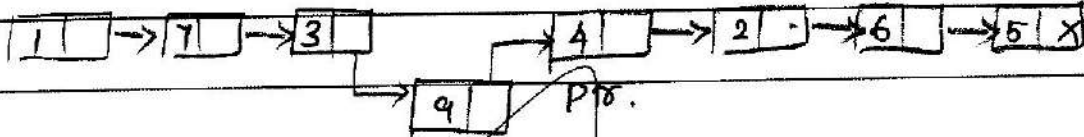
9

new_node

Take two pointer variables ptr and $preptr$ and initialize them with $start$ so that $start$, ptr , and $preptr$ point to the first node of the list.



Add the new node in between the node pointed by preptr and ptr



Start .

20/20

good
KOD



ACADEMIC YEAR 2019-20 (ODD SEMESTER)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CS8391 – DATA STRUCTURES**

PCE SKILL ACTIVITY PLAN

Activity 1: Application of concept -Consider five cities: (1) New Delhi, (2) Mumbai, (3) Chennai, (4) Bangalore, and (5) Kolkata, and a list of flights that connect these cities as shown in the following table. Use the given information to construct a graph and explain.(10)

Flight No.	Origin	Destination
101	2	3
102	3	2
103	5	3
104	3	4
105	2	5
106	5	2
107	5	1
108	1	4
109	5	4
110	4	5

Activity 2: GATE questionnaire*

Activity 3: Concept Map – Sorting Algorithm

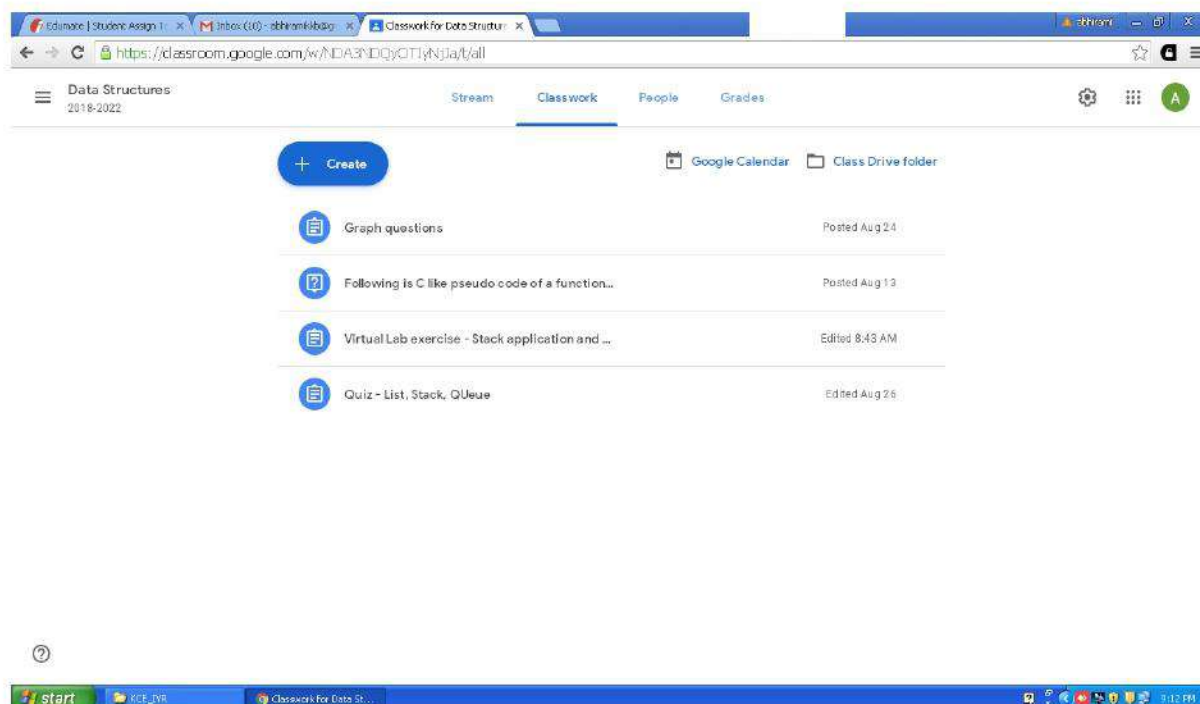
Activity 4: Virtual Lab exercise – Expression trees

Activity 5: Coding exercise – Trees & Graphs

Activity 6: Google classroom – Quiz*

PCE SKILL ACTIVITY EXECUTION –GOOGLE CLASSROOM BASED

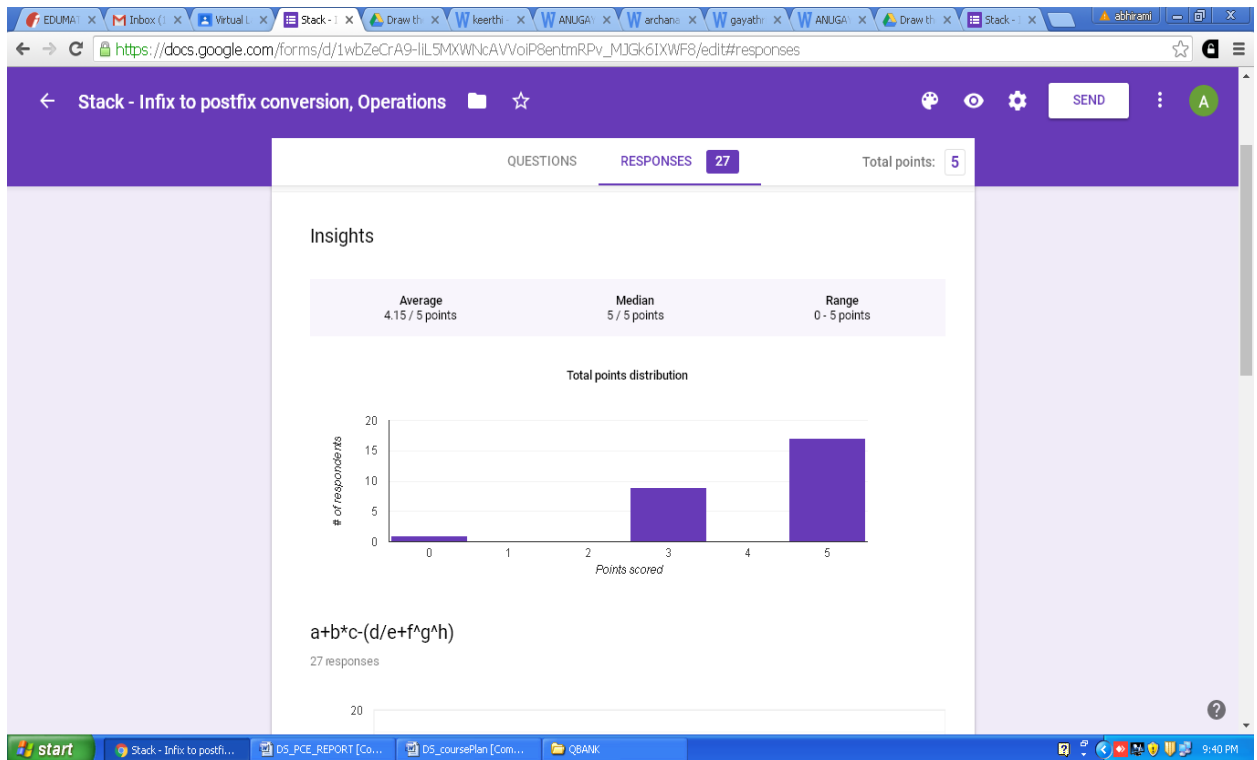
(17.8.19 & 24.8.19)



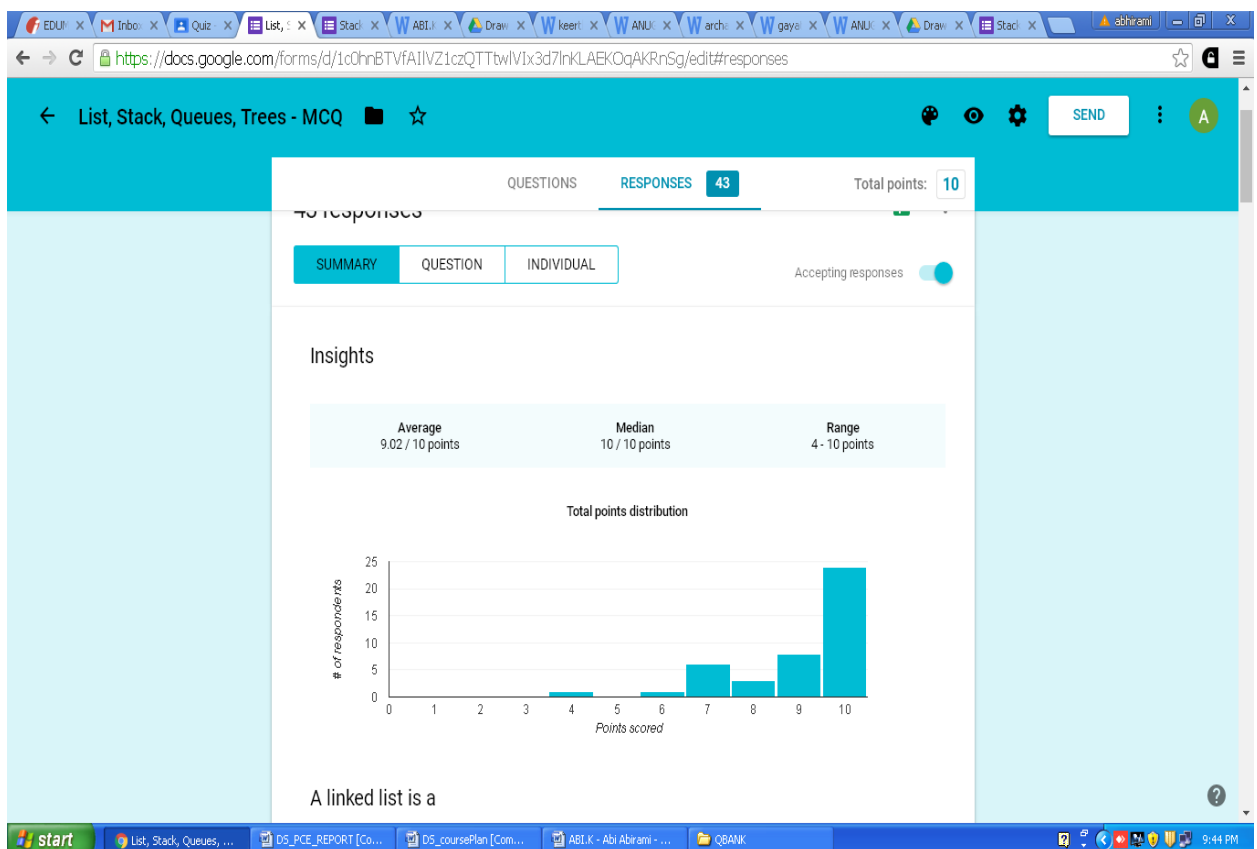
PCE SKILL ACTIVITY EXECUTION SUMMARY

ACTIVITY	MODE	OUTCOME	MARKS ALLOTTED	NO. OF STUDENTS APPEARED
Application of Graph Concept No. of Questions: 8	Written (Advanced Learners)	Ability to apply graph concepts to solve real world problem	5	09
GATE Questionnaire - quiz No. of Questions: 10	Online	Ability to gain indepth knowledge and become competent for GATE exams	5	45*
Virtual Lab exercise No. of question:2	Online	Ability to convert arithmetic expressions into different notations Gain skills to apply Stack data structure and solve problems	10	45*
Coding Exercise No. of questions:1	Online	Ability to trace programming exercise, use data structure and find output	5	45*
Total Marks (appeared by all students)			20	-

SCREEN SHOTS – ACTIVITY 2



Activity 2- (quiz – average marks 9.02/10)- 0.5 weigthage



SCREEN SHOTS – ACTIVITY 4 (coding exercise)- 5 marks weigthage

Data Structures
2018-2022

Question | **Student answers**

Return | 5 points

All students

Sort by status

Assigned

Student	Turned in	Assigned	Graded
Abirami Krishnamoorthi	0	14	38
ARCHANA ANBALAGAN			
Gandarvakottai Bdo			
Gopi Nathan			
Jemima O7			

Following is C like pseudo code of a function that takes a number as an argument, and uses a stack S to do processing. void fun(int n) { Stack S; // Say it creates an empty stack S while (n > 0) { // This line pushes the value of n%2 to stack S push(&S, n%2); n = n/2; } // Run while Stack S is not empty while (!isEmpty(&S)) printf("%d ", pop(&S)); // pop an element from S and print it } What does the above function do in general?

Answer	Score
Prints binary representation of n in reverse order	29
Prints binary representation of n	6
Prints the value of Logn	3
Prints the value of Logn in reverse order	0

SCREEN SHOTS – ACTIVITY 3 (Virtual lab exercise)- 5 marks weigthage

Stack - Infix to postfix conversion, Operations

The name and photo associated with your Google account will be recorded when you upload files and submit this form. Not [abhiramikb@gmail.com](#)? [Switch account](#)

*** Required**

a+b*c-(d/e+f^g^h) * 2 points

Your answer

Draw the stack structure in each case when the following operations are performed on an empty stack. (a) Add A, B, C, D, E, F (b) Delete two letters (c) Add G (d) Add H (e) Delete four letters (f) Add I * 3 points

[ADD FILE](#)

SUBMIT

SCREEN SHOTS – ACTIVITY 3 (infix to postfix conversion)- 2 marks weighthage

0	a	(	a
1	+	(+	a
2	b	(+	ab
3	*	(+*	ab
4	c	(+*	abc
5	-	(-	abc*+
6	(	(- (	abc*+
7	d	(- (	abc*+d
8	/	(- (/	abc*+d
9	e	(- (/	abc*+de
10	+	(- (+	abc*+de/
11	f	(- (+	abc*+de/f
12	^	(- (+ ^	abc*+de/f
13	g	(- (+ ^	abc*+de/fg
14	^	(- (+ ^ ^	abc*+de/fg
15	h	(- (+ ^ ^	abc*+de/fgh
16	)	(-	abc*+de/fgh ^ ^ +
17		(-	abc*+de/fgh ^ ^ +
18	)		abc*+de/fgh ^ ^ + -

SCREEN SHOTS – ACTIVITY 3 – online assignment submission based

Stack - Infix to postfix conversion, Operations

QUESTIONS RESPONSES 27 Total points: 5

performed on an empty stack.(a) Add A, B, C, D, E, F (b) Delete two letters(c) Add G (d) Add H(e) Delete four letters (f) Add I

27 responses

- postfix - Kaviyan Editz.docx
- ABI K - Abi Abirami.docx
- a - kaman kar.docx
- archana.a - Archana Archana.docx
- gayathri - Jegan Jegan.docx
- ANUGAYA - Gayathri .K.docx
- ABI - P Abirami.docx
- STACK - aburvanayagi thirumurgan.docx
- PCE - guna manza.docx
- keerthi - Keerthana Venkatesan.docx

17 more files

Files stored in Drive VIEW FOLDER

SCREEN SHOTS – ACTIVITY 3 – sample assignment

Draw the stack structure in each case when the following operations are performed on an empty stack. (a) Add A, B, C, D, E, F (b) Delete two letters (c) Add G (d) Add H (e) Delete four letters (f) Add I

F
E
D
C
B
A

STEP 1

D
C
B
A

STEP 2

G
D
C
B
A

STEP 3

H
G
D
C
B
A

STEP 4

B
A

STEP 5

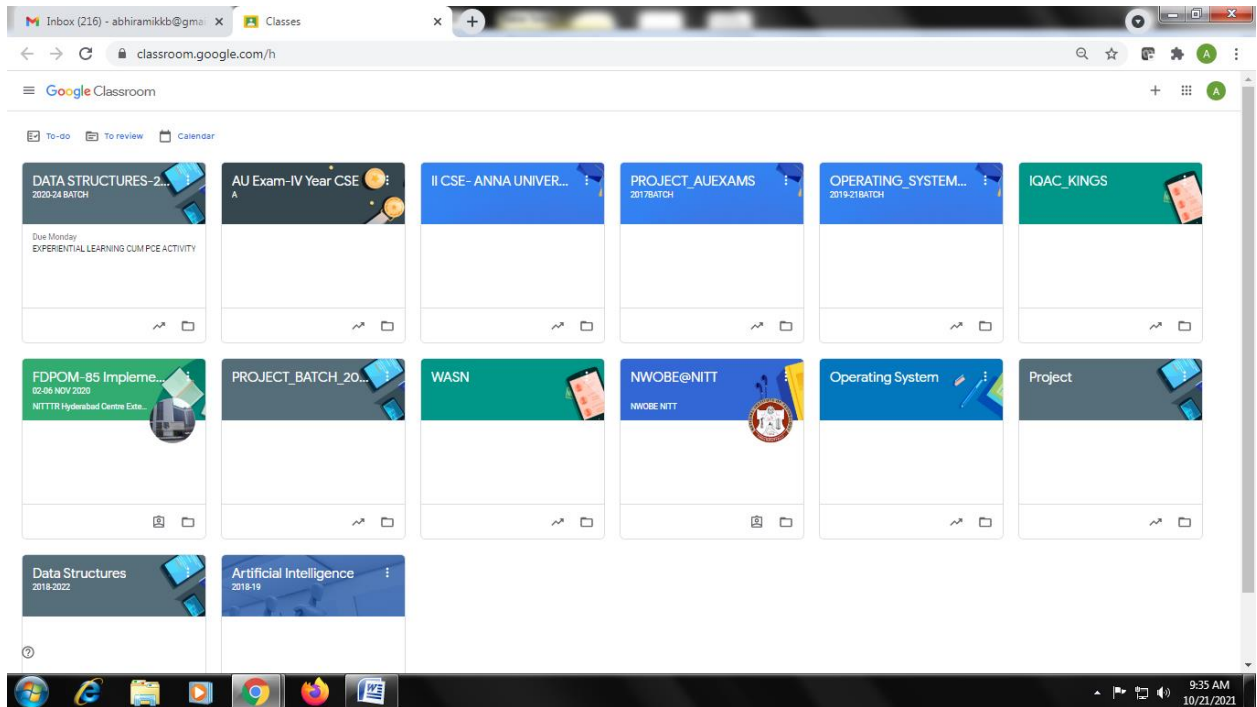
I
B
A

STEP 6

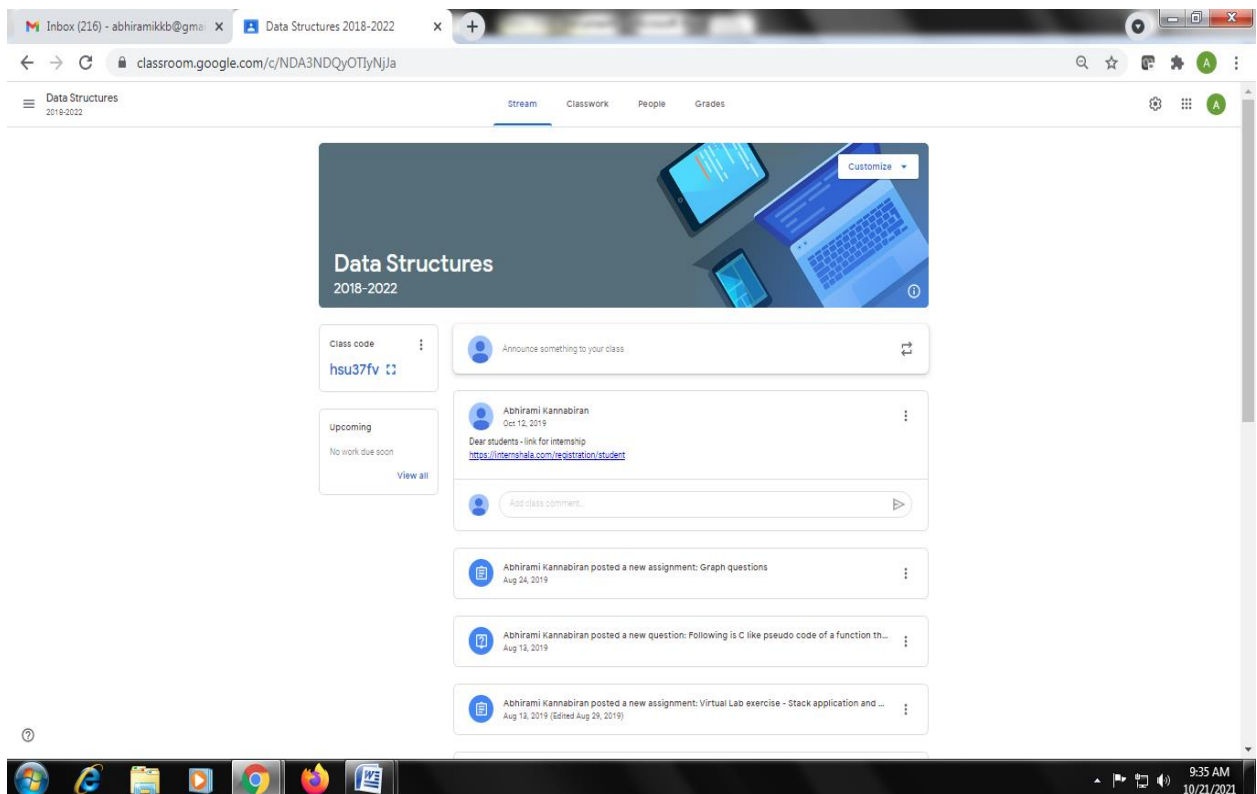
PCE SKILL ACTIVITY MARK ALLOCATION

S.NO	REGISTER NO	NAME	A1(5)	A2(5)	A3(10)	A4(5)	TOTAL
1.	821118104001	Abarna M		5	10	5	20
2.	821118104002	Abinaya N		5	10	5	20
3.	821118104003	Abirami K	5	5	10	5	20
4.	821118104004	Abirami M	5	5	10	5	20
5.	821118104005	Abirami P		5	10	5	20
6.	821118104006	Aburvanayaki T		5	10	5	20
7.	821118104007	Aiswarya K		5	10	5	20
8.	821118104008	Archana A		5	10	5	20
9.	821118104009	Arunkumar G		5	10	5	20
10.	821118104010	Babu M		5	10	5	20
11.	821118104011	Deepika V		5	10	5	20
12.	821118104012	Dharshini T R		5	10	5	20
13.	821118104013	Divyabharathi G	5	5	10	5	20
14.	821118104014	Gayathri J		5	10	5	20
15.	821118104015	Gayathri K	5	5	10	5	20
16.	821118104016	Gunaseelan B		5	10	5	20
17.	821118104017	Hariharan E		5	10	5	20
18.	821118104018	Iswarya S		5	10	5	20
19.	821118104019	Jemima Esther Grace M	5	5	10	5	20
20.	821118104020	Kamalapriya M		5	10	5	20
21.	821118104021	Kaviyachelvan K		5	10	5	20
22.	821118104022	Keerthana V		5	10	5	20
23.	821118104024	Nandha kishore S		5	10	5	20
24.	821118104025	Narsrin banu N		5	10	5	20
25.	821118104026	Nathiya devi N		5	10	5	20
26.	821118104027	Naveen sundhar V		5	10	5	20
27.	821118104028	Pathmanaban R		5	10	5	20
28.	821118104029	Priya G		5	10	5	20
29.	821118104030	Priyadharshini M		5	10	5	20
30.	821118104031	Priyadharshini R	5	5	10	5	20
31.	821118104032	Ramya J		5	10	5	20
32.	821118104033	Rathnakumar T		5	10	5	20
33.	821118104034	Sarathkumar R		5	10	5	20
34.	821118104035	Selvarani T	5	5	10	5	20
35.	821118104036	Sivaranjani T		5	10	5	20
36.	821118104037	Sneha K		5	10	5	20
37.	821118104038	Sneka S		5	10	5	20
38.	821118104039	Sowmiya K		5	10	5	20
39.	821118104040	Srinithi C		5	10	5	20
40.	821118104041	Suruthi S	5	5	10	5	20
41.	821118104042	Surya R		5	10	5	20
42.	821118104043	Vaishnavi P		5	10	5	20
43.	821118104044	Vithyatharan S		5	10	5	20
44.		Gopinathan.D		5	10	5	20
45.		Sindu.P		5	10	5	20

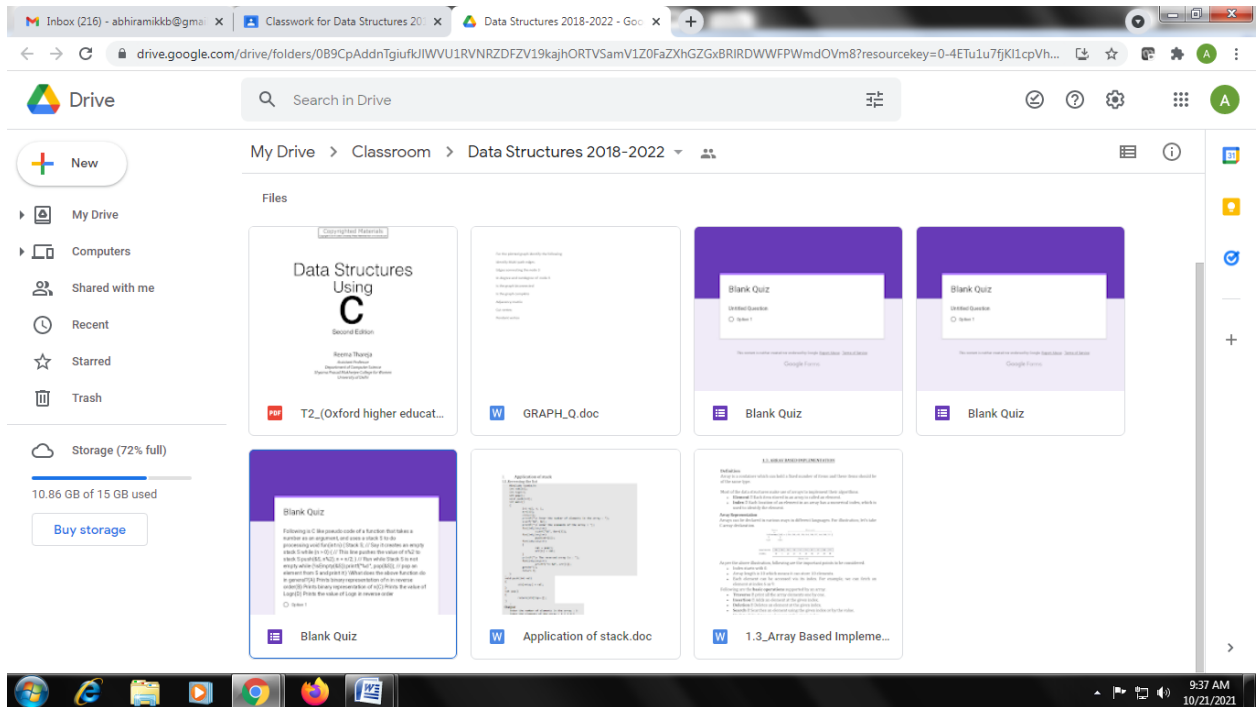
STAFF GOOGLE CLASSROOM - SNAPSHOT



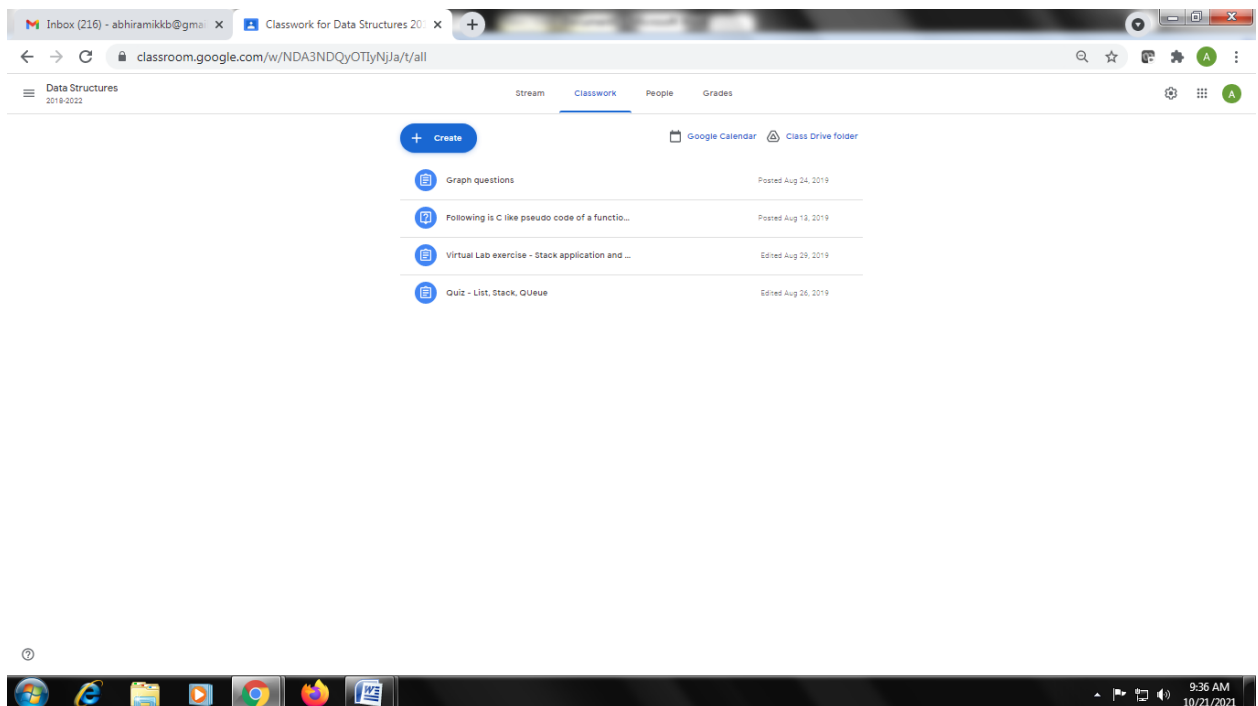
COURSE CLASSROOM SNAPSHOT



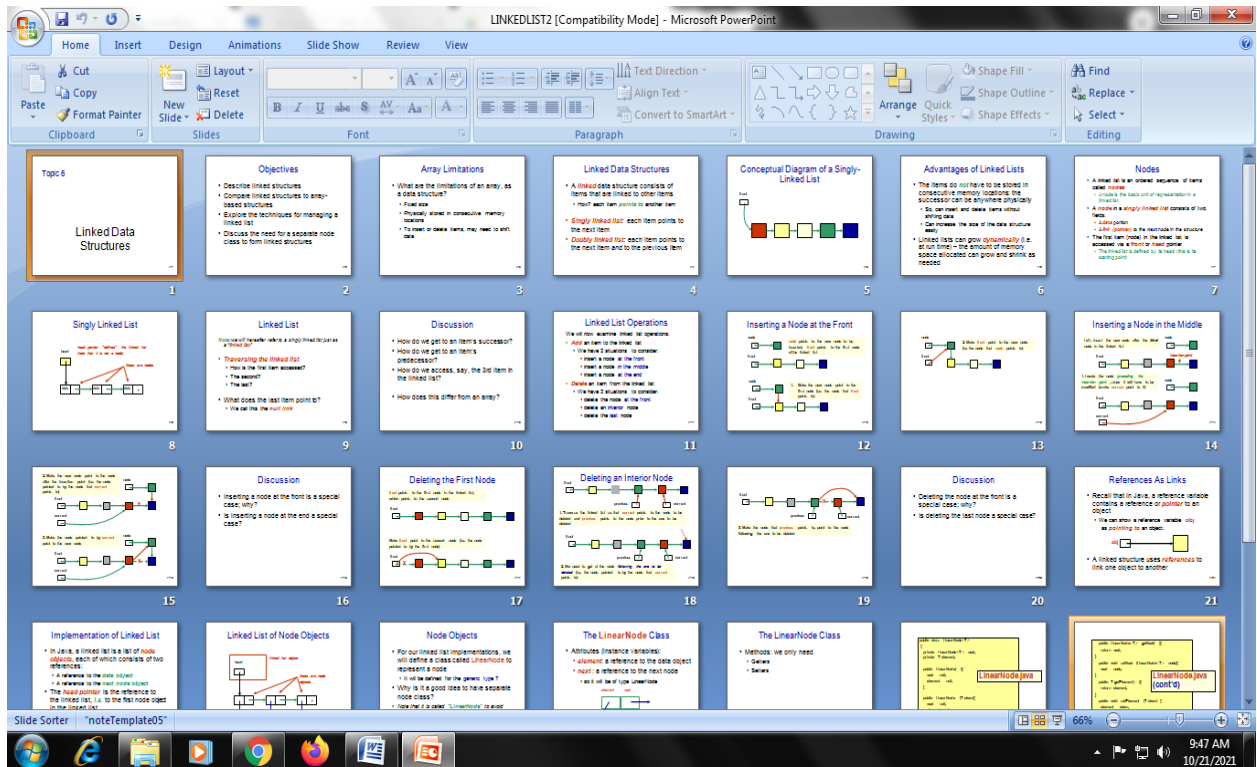
COURSE DRIVE SNAPSHOT



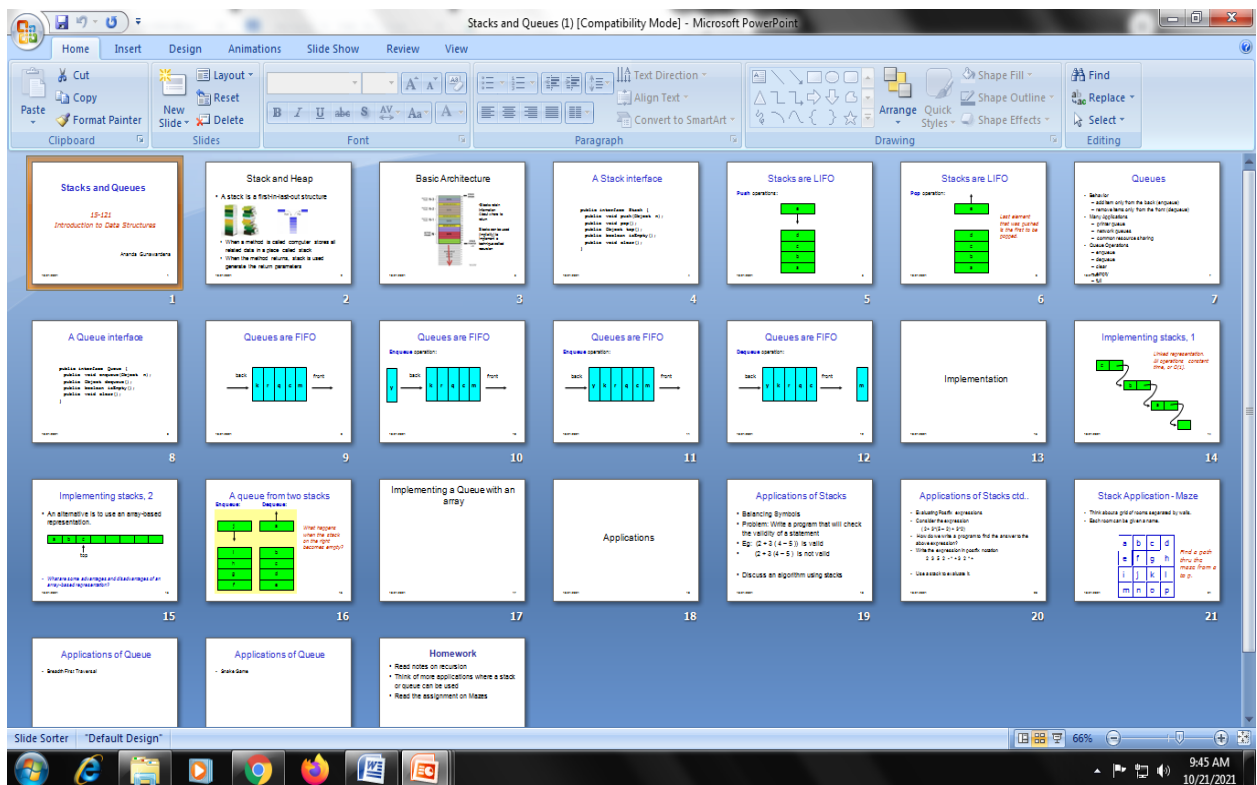
COURSE ACTIVITY ASSIGNED - SNAPSHOT



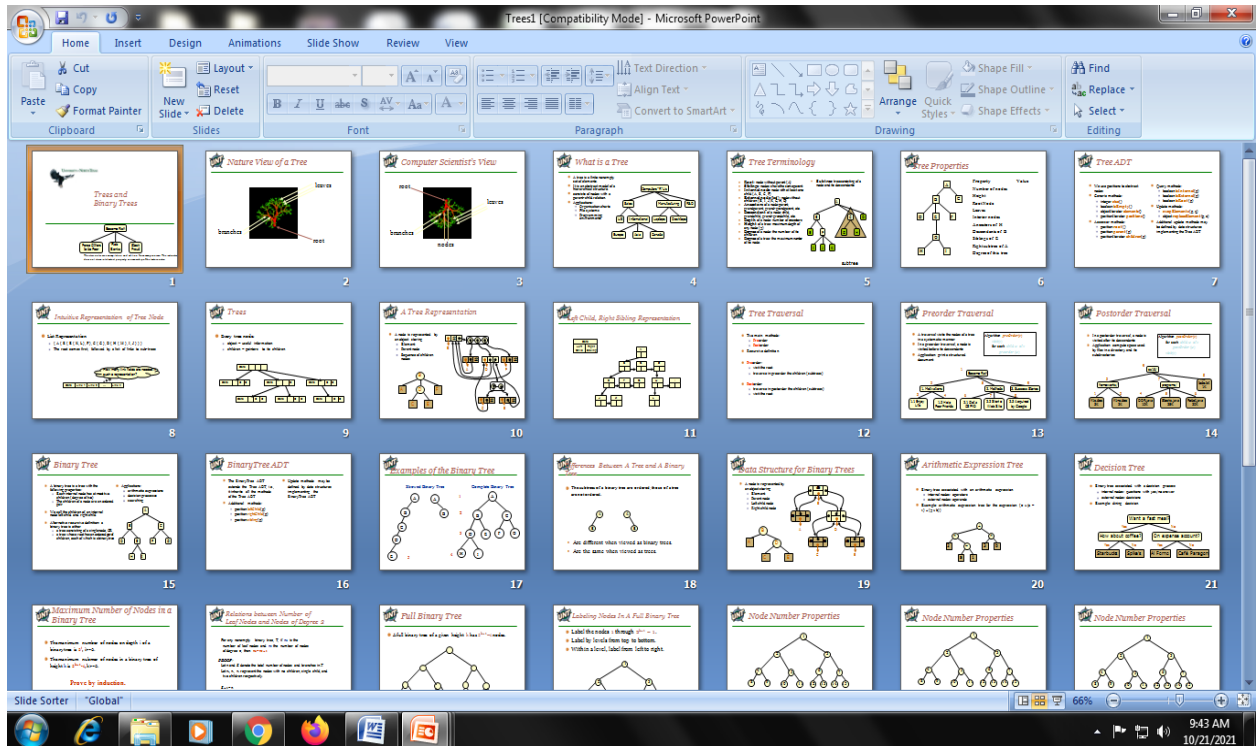
UNIT-1-SAMPLE PPT – SLIDE SORTER VIEW SNAPSHOT



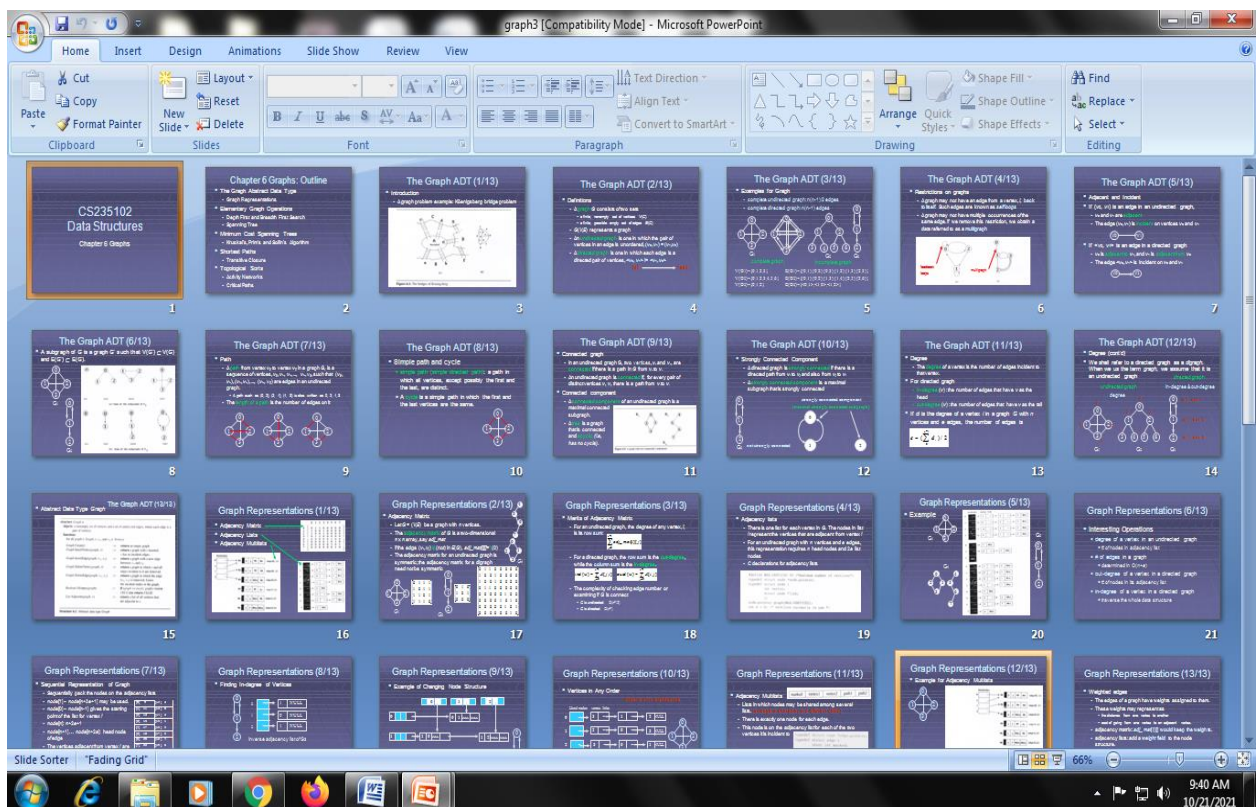
UNIT-2-SAMPLE PPT – SLIDE SORTER VIEW SNAPSHOT



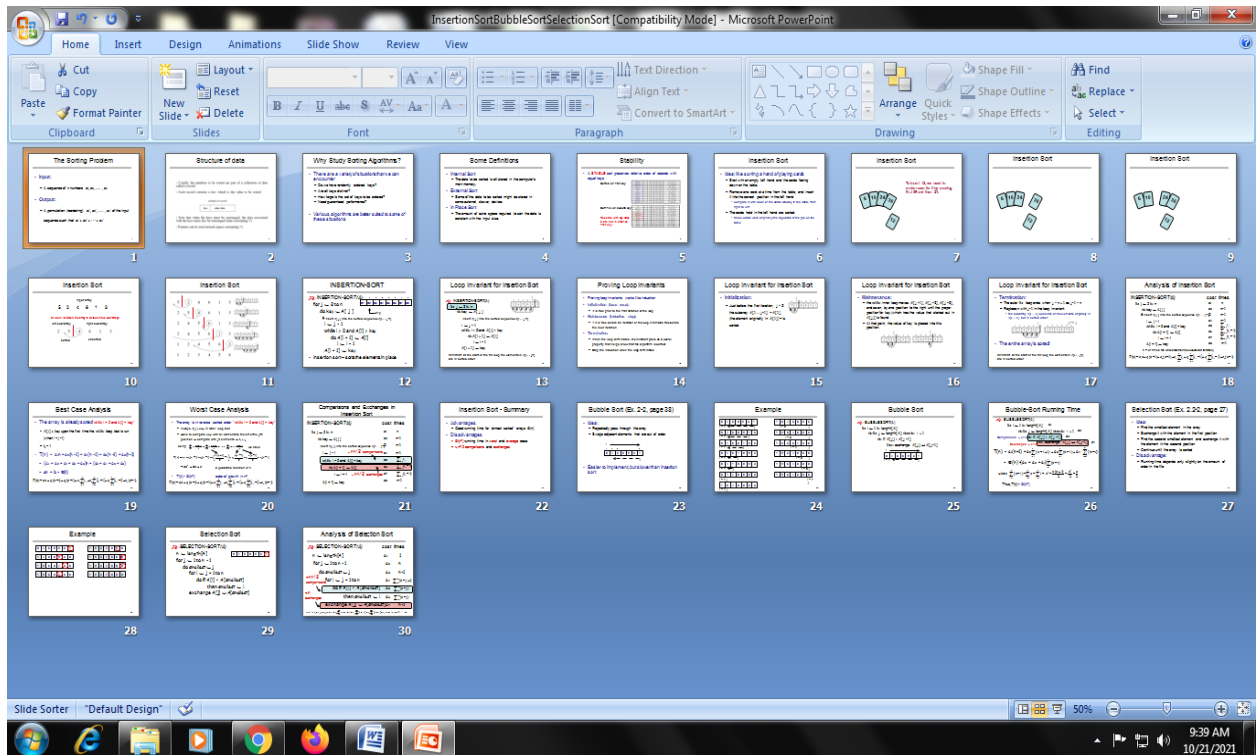
UNIT-3-SAMPLE PPT - SLIDE SORTER VIEW SNAPSHOT



UNIT-4-SAMPLE PPT - SLIDE SORTER VIEW SNAPSHOT



UNIT-5-SAMPLE PPT – SLIDE SORTER VIEW SNAPSHOT



NPTEL SESSION ON TREE TRAVERSAL

Inbox (216) - abhiramikkb@gmail.com | NPTEL : Computer Science and Engineering

npTEL.ac.in/courses/106/102/106102064/

NPTEL About us Courses Contact us

Courses Computer Science and Engineering Data Structures And Algorithms (Video) **Syllabus** Co-ordinated by : IIT Delhi

Available from : 2009-12-31 Lec : 7

Modules / Lectures

- Data Structures And Algorithms
 - Introduction to Data Structures and Algorithms
 - Stacks
 - Queues and Linked Lists
 - Dictionaries
 - Hashing
 - Trees
 - Tree Walks / Traversals
 - Ordered Dictionaries
 - Deletion

Watch on YouTube Transcripts Books

Lecture - 7 Tree Walks / Traversals Watch later Share

```
public abstract class BinaryTreeTraversal {
    protected BinaryTreeNode root;

    protected BinaryTreeNode rootNode() {
        return root;
    }

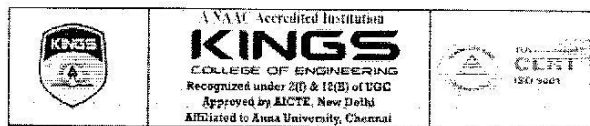
    protected BinaryTreeNode leftChild() {
        return leftChild;
    }

    protected BinaryTreeNode rightChild() {
        return rightChild;
    }

    return result;
}
```

Watch on YouTube

9:58 AM 10/21/2021



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Course File

Format B

CONTENT BEYOND THE SYLLABUS

TITLE	: Applications of Graphs- Virtual Lab Session Spanning Trees- Minimum Spanning Tree , Prim's Algorithm, Kruskal Algorithm , Dijitra's Algorithm
OBJECTIVE	: To Obtain skills in identifying shortest path for the given graph
METHODOLOGY	: Demonstration cum problem solving Exercise
EVALUATION	: Test on the above exercise included in Revision, Model
DATE OF COMPLETION	: 18.9.2019

K. S. Sree
Staff In-charge 18/9/19

CONTENT BEYOND SYLLABUS

MINIMUM SPANNING TREE

SAMPLE SCREENSHOT

The screenshot shows the Virtual Labs website interface. The browser tabs include 'Inbox (219) - abhiramikkb@gmail.com', 'NEW HORIZON COLLEGE OF ENGINEERING', 'Virtual Labs - Computer Science', and 'Virtual Labs'. The address bar shows the URL 'cse01-iiith.vlabs.ac.in/exp/minimum-spanning-trees/'. The website header includes the Virtual Labs logo and navigation links for HOME, PARTNERS, and CONTACT. The main content area is titled 'Computer Science and Engineering > Data Structures > Experiments' and 'Spanning Trees in Graphs'. A sidebar on the left lists navigation options: Aim, Objective, Theory, Simulation, Assignment, and Feedback. The main text describes the experiment: 'In this experiment, we will see a famous problem in graphs, finding the Minimum Spanning Tree. Let $G = (V, E, W)$ be a weighted graph. Find a subgraph G' of G that is connected and has the smallest cost, where cost is defined as the sum of the edge weights of all edges in G' . Observe that if G' has a cycle, we can remove one of the edges along a cycle and still the resultant graph will remain connected and has smaller cost than G' . Hence, G' cannot have a cycle and as it has to be connected, G' must be a tree. We define G' as a spanning subgraph of G iff $V(G) = V(G')$ and a spanning subgraph that is also a tree is called a spanning tree of G . Our aim is to find a spanning tree of G that has the least cost and such a spanning tree is called as minimum spanning tree (MST) of G .' The footer includes 'Community Links' (Sakshat Portal, Outreach Portal, FAQ: Virtual Labs), 'Contact Us' (Phone: 011-26582050, Email: support@vlabs.ac.in), and 'Follow Us' (Twitter, Facebook, YouTube, LinkedIn). The license is 'AGPL 3.0 & Creative Commons (CC BY-NC-SA 4.0)'.

The screenshot shows the Virtual Labs website interface for the 'Kruskals Algorithm' demo. The browser tabs include 'Inbox (219) - abhiramikkb@gmail.com', 'NEW HORIZON COLLEGE OF ENGINEERING', 'Virtual Labs - Computer Science', and 'Virtual Labs'. The address bar shows the URL 'ds2-iiith.vlabs.ac.in/exp/min-spanning-trees/kruskals-algorithm/mst_kruskal_demo.html'. The website header includes the Virtual Labs logo and navigation links for HOME, PARTNERS, and CONTACT. The main content area is titled 'Instructions' and shows a graph with nodes a, b, c, d, e, and f. The graph edges and weights are: (b,d) weight 8, (b,c) weight 11, (c,d) weight 12, (c,a) weight 13, (d,e) weight 16, (a,e) weight 15, and (e,f) weight 10. A legend indicates: 'Not Visited' (black line), 'Edge added to MST' (red line), 'Edge discarded' (grey line), and 'Edge being checked' (blue line). A 'Min Speed' slider is set to 'Max Speed'. The 'Observations' section shows 'Checking smallest edge among unchecked edges - 8'. The bottom of the interface has buttons for 'Next', 'New Graph', 'Reset', and 'Pause'. The sidebar on the left lists navigation options: Recap: Algorithm, MST, Kruskals Algorithm, Aim, Concept, Algorithm, Demo, Practice, Exercise, Analysis of Kruskals, Prims Algorithm, Analysis of Prims, Further Readings/References, and Feedback.

Recap: Algorithm

- MST
- Kruskals Algorithm
- Analysis of Kruskals
- Prims Algorithm
- Aim
- Concept
- Algorithm
- Demo
- Practice
- Exercise
- Analysis of Prims
- Further Readings/References
- Feedback

Instructions

```

graph TD
    a ---|13| b
    a ---|15| e
    b ---|11| c
    b ---|8| d
    c ---|12| d
    d ---|16| e
    b ---|10| e
  
```

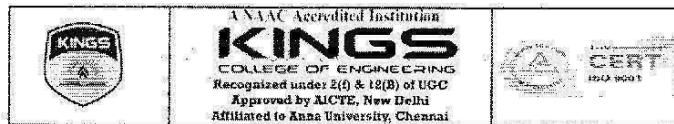
Min Speed Max Speed

Observations

Starting from node 'b' edges from the node are added to edges to be checked

Next New Graph Reset Pause

— Not Visited
 — Edges to be checked
 — Added to MST
 — Edge discarded
 — Edge being checked



General Practice Followed

1. Class test (Part-A Qns) - Notebook
2. Utilization of DS lab to support theoretical aspects of DS
3. Assessment – Prior day coaching
4. Unitwise topic list & coverage tracking
No Repetition of questions in Class Test, Assessment Tests
5. Suitable ppt session & Google classroom usage
6. Practice on book exercises
7. Assessment feedback on performance

Slow Learner , Average Learner

1. Evening coaching classes (common slot)
2. Revision , ICC classes (common slot)

K. Colee 28/6/19
Staff In-charge



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ACADEMIC YEAR 2019-20 / ODD SEMESTER
II CSE NAME LIST-Advanced Learner

S.NO	REGISTER NO	NAME
1.	821118104001	Abarna M
2.	821118104003	Abirami K
3.	821118104005	Abirami P
4.	821118104007	Aiswarya K
5.	821118104008	Archana A
6.	821118104009	Arunkumar G
7.	821118104011	Deepika V
8.	821118104013	Divyabharathi G
9.	821118104014	Gayathri J
10.	821118104015	Gayathri K
11.	821118104016	Gunaseelan B
12.	821118104017	Hariharan E
13.	821118104024	Nandha kishore S
14.	821118104026	Nathiya devi N
15.	821118104033	Rathnakumar T
16.	821118104035	Selvarani T
17.	821118104040	Srinithi C
18.	821118104041	Suruthi S
19.	821118104043	Vaishnavi P
20.	821118104044	Vithyatharan S

Staff Signature



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING ACADEMIC YEAR 2019-20 / ODD SEMESTER
II CSE NAME LIST-Average Learner**

S.NO	REGISTER NO	NAME
1.	821118104006	Aburvanayaki T
2.	821118104012	Dharshini T R
3.	821118104019	Jemima Esther Grace M
4.	821118104020	Kamalapriya M
5.	821118104022	Keerthana V
6.	821118104025	Narsrin banu N
7.	821118104028	Pathmanaban R
8.	821118104031	Priyadharshini R
9.	821118104032	Ramya J
10.	821118104034	Sarathkumar R
11.	821118104036	Sivaranjani T
12.	821118104038	Sneka S
13.	821118104039	Sowmiya K
14.	821118104042	Surya R
15.		Gopinathan.D
16.		Sindu.P

Staff Signature



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING ACADEMIC YEAR 2019-20 / ODD SEMESTER
II CSE NAME LIST- Slow Learner**

S.NO	REGISTER NO	NAME
1.	821118104002	Abinaya N
2.	821118104004	Abirami M
3.	821118104010	Babu M
4.	821118104018	Iswarya S
5.	821118104021	Kaviyachelvan K
6.	821118104027	Naveen sundhar V
7.	821118104029	Priya G
8.	821118104030	Priyadharshini M
9.	821118104037	Sneha K

Staff Signature



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ACADEMIC YEAR 2019 - 2020 / ODD SEMESTER

Guidelines for Remedial class

- It is planned to conduct the Phase 1 coaching class to support the students who scored less than 16 out of 20 in weektest-I during 4.40-5.30 pm. They will write retest during that time.
- Phase 2 coaching class will be conducted for the students those who secure less than 16 out of 20 in weektest-II during 4.40-5.30 pm and they have to write retest.
- Students those who secured Less than 70 marks in assessment II will attend the coaching classes on Saturdays. They have to learn and write test during coaching.
- Special coaching will be given to the students except all clear students during afternoon session of model examination.
- Intensive coaching class will be conducted from 21.10.19-1.11.19

[Handwritten Signature]
5/7/2019
H.O.O/LSC



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ACADEMIC YEAR 2019 - 20 ODD SEMESTER
SATURDAY SPECIAL TIME TABLE - 13.07.19

Session	1	2	10.45 am - 11.00 am	3	4	12.30 pm - 01.15 pm	5	6	02.45 pm - 03.00 pm	7	8
Day	09.15am - 10.00am	10.00am - 10.45am		11.00am - 11.45am	11.45am - 12.30pm		01.15pm - 02.00pm	02.00pm - 02.45pm		03.00pm - 03.45pm	03.45pm - 04.30pm
II CSE	CS8391	CS8392	BREAK	CS8351		LUNCH BREAK	EC8395		BREAK	MA8351	
III CSE	EC8691	OMF551		CS8591	OMF551		IIT BOMBAY TUTORIAL			MA8551	CS8592
IV CSE	CS6704	CS6703		CS6004	CS6701		CS6007	CS6702		IIT BOMBAY TUTORIAL	

SUB CODE	NAME OF THE SUBJECT	NAME OF THE STAFF
II CSE		
MA8351	Discrete Mathematics	Ms.S.Geetha
CS8351	Digital Principles and System Design	Mr.S.Rajarajan
CS8391	Data Structures	Ms.K.Abhirami
CS8392	Object Oriented Programming	Dr.D.Sivakumar
EC8395	Communication Engineering	Mr.T.Jeyaseelan
III CSE		
MA8551	Algebra and Number Theory	Mr.G.Jeyakrishnan
CS8591	Computer Networks	Mr.K.Rajesh
EC8691	Microprocessor & Microcontroller	Dr.S.M.Uma
CS8592	Object Oriented Analysis & Design	Ms.R.Ranitha
OMF551	Product Design and Development	Ms.R.Suganthalakshmi
	IIT BOMBAY TUTORIAL	Ms.K.Abhirami
IV CSE		
CS6701	Cryptography and Network Security	Dr.D.Sivakumar
CS6702	Graph Theory and Applications	Dr.R.Suresh
CS6703	Grid and Cloud Computing	Ms.B.Sangeetha
CS6704	Resource Management Techniques	Dr.G.Shankara Kalidoss
CS6004	Cyber Forensics	Mr.R.Sriramkumar
	IIT BOMBAY TUTORIAL	Mr.R.Sriramkumar

W. S. P.
CLASS INCHARGE

S. S.
HOD/CSE



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ACADEMIC YEAR 2019 – 2020 (ODD SEMESTER)
COACHING CLASS SCHEDULE

Year	Date	Sub Code	Subject Name	Staff Name
II	05.08.19	MA8351	Discrete Mathematics	Ms.S.Geetha
	06.08.19	CS8351	Digital Principles and System Design	Mr.S.Rajarajan
	07.08.19	CS8391	Data Structures	Ms.K.Abhirami
	08.08.19	CS8392	Object Oriented Programming	Dr.D.Sivakumar
	09.08.19	EC8395	Communication Engineering	Mr.T.Jeyaseelan
III	05.08.19	MA8551	Algebra and Number Theory	Mr.G.Jeyakrishnan
	06.08.19	CS8591	Computer Networks	Mr.K.Rajesh
	07.08.19	EC8691	Microprocessor & Microcontroller	Dr.S.M.Uma
	08.08.19	CS8501	Theory of Computation	Ms.S.Puvaneswari
	09.08.19	CS8592	Object Oriented Analysis & Design	Ms.R.Ranitha
	13.08.19	OMF551	Product Design and Development	Ms.R.Suganthalakshmi
IV	05.08.19	CS6701	Cryptography and Network Security	Dr.D.Sivakumar
	06.08.19	CS6702	Graph Theory and Applications	Dr.R.Suresh
	07.08.19	CS6703	Grid and Cloud Computing	Ms.B.Sangeetha
	08.08.19	CS6704	Resource Management Techniques	Dr.G.Shankara Kalidoss
	09.08.19	CS6004	Cyber Forensics	Mr.R.Sriramkumar
	13.08.19	CS6007	Information Retrieval	Ms.P.Nalayini

[Signature]
HOD/CSE



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ACADEMIC YEAR 2019 – 2020 (ODD SEMESTER)
ASSESSMENT TEST – II REVISION CLASS SCHEDULE

Year	Date	Sub Code	Subject Name	Staff Name
II	03.09.19	MA8351	Discrete Mathematics	Ms.S.Geetha & Mr.S.Rajaraman
	04.09.19	CS8351	Digital Principles and System Design	Mr.S.Rajaraman & Ms.K.Abhirami
	05.09.19	CS8391	Data Structures	Ms.K.Abhirami & Mr.S.Rajaraman
	06.09.19	CS8392	Object Oriented Programming	Dr.D.Sivakumar & Mr.T.Jeyaseelan
	07.09.19	EC8395	Communication Engineering	Mr.T.Jeyaseelan & Dr.D.Sivakumar
III	04.09.19	CS8591	Computer Networks	Mr.K.Rajesh & Ms.R.Ranitha
	05.09.19	EC8691	Microprocessor & Microcontroller	Dr.S.M.Uma & Mr.K.Rajesh
	06.09.19	CS8501	Theory of Computation	Ms.S.Puvaneswari & Ms.R.Suganthalakshmi
	07.09.19	CS8592	Object Oriented Analysis & Design	Ms.R.Ranitha & Ms.S.Puvaneswari
	09.09.19	OMF551	Product Design and Development	Ms.R.Suganthalakshmi & Ms.S.Puvaneswari
IV	03.09.19	CS6701	Cryptography and Network Security	Dr.D.Sivakumar & Mr.R.Sriramkumar
	04.09.19	CS6702	Graph Theory and Applications	Dr.R.Suresh & Dr.G.Shankara Kalidoss
	05.09.19	CS6703	Grid and Cloud Computing	Ms.B.Sangeetha & Ms.P.Nalayini
	06.09.19	CS6704	Resource Management Techniques	Dr.G.Shankara Kalidoss & Dr.R.Suresh
	07.09.19	CS6004	Cyber Forensics	Mr.R.Sriramkumar & Dr.D.Sivakumar
	09.09.19	CS6007	Information Retrieval	Ms.P.Nalayini & Ms.B.Sangeetha

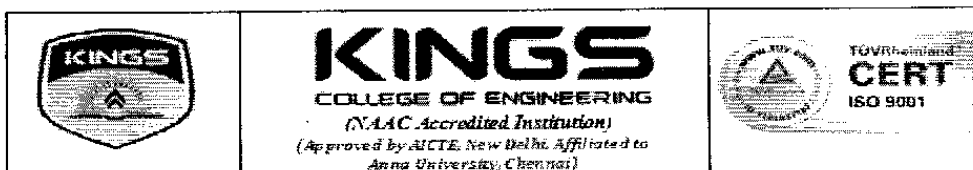
II CSE – *[Signature]*

III CSE – *[Signature]*

IV CSE – *[Signature]*

CLASS INCHARGE

[Signature] 24/9/19
HOD/CSE



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ACADEMIC YEAR 2019 - 20 ODD SEMESTER
SATURDAY COACHING CLASS TIME TABLE**

Session	DATE	1	2	10.45 am	3	4	12.30 pm	5	6	02.45 pm	7	8
Day		09.15am - 10.00am	10.00am - 10.45am	- 11.00 am	11.00am - 11.45am	11.45am - 12.30pm	- 01.15 pm	01.15pm - 02.00pm	02.00pm - 02.45pm	- 03.00 pm	03.00pm - 03.45pm	03.45pm - 04.30pm
I CSE	14.09.19	CS8391	CS8351	BREAK	CS8392		LUNCH BREAK	EC8395		BREAK	MA8351	
	28.09.19	CS8391			CS8351			EC8395			CS8392	
II CSE	14.09.19	CS8501			CS8501	EC8691		EC8691			MA8551	
	28.09.19	CS8591			CS8591	OMF551		OMF551			CS8592	
IV CSE	14.09.19	CS6703			CS6007			CS6004			CS6701	
	28.09.19	CS6704			CS6701			CS6702			CS6703	CS6704

SUB CODE	NAME OF THE SUBJECT	NAME OF THE STAFF
II CSE		
MA8351	Discrete Mathematics	Ms.S.Geetha
CS8351	Digital Principles and System Design	Mr.S.Rajarajan
CS8391	Data Structures	Ms.K.Abhirami
CS8392	Object Oriented Programming	Dr.D.Sivakumar
EC8395	Communication Engineering	Mr.T.Jeyaseelan
III CSE		
MA8551	Algebra and Number Theory	Mr.G.Jeyakrishnan
CS8591	Computer Networks	Mr.K.Rajesh
EC8691	Microprocessor & Microcontroller	Dr.S.M.Uma
CS8592	Object Oriented Analysis & Design	Ms.R.Ranitha
OMF551	Product Design and Development	Ms.R.Suganthalakshmi
CS8501	Theory of Computation	Ms.S.Puvaneswari
IV CSE		
CS6701	Cryptography and Network Security	Dr.D.Sivakumar
CS6702	Graph Theory and Applications	Dr.R.Suresh
CS6703	Grid and Cloud Computing	Ms.B.Sangeetha
CS6704	Resource Management Techniques	Dr.G.Shankara Kalidoss
CS6004	Cyber Forensics	Mr.R.Sriramkumar
CS6007	Information Retrieval Techniques	Ms.P.Nalayini

W. Cse - S. Puv
CLASS INCHARGE

S. S. 12/9/19
HOD/CSE



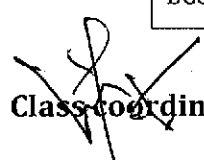
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ACADEMIC YEAR 2019-2020 (ODD SEMESTER)
ICC- TIME TABLE with effect from 16.10.2019

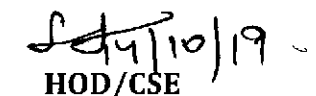
Year / Sem : II/ III

S.NO	DATE	SUBJECT CODE & NAME (FN)		SUBJECT CODE & NAME (AN)	
		9.15 a.m. to 12.30 p.m.		1. 15 p. m to 4.30p.m	
		Hall No.222(1-22)	Hall No. 227 (23-45)	Hall No.222(1-22)	Hall No. 227 (23-45)
1.	16.10.19 WEDNESDAY	MA8351 - Discrete Mathematics		CS8351 - Digital Principles and System Design	CS8391 - Data Structures
2.	17.10.19 THURSDAY	CS8351 - Digital Principles and System Design	CS8391 - Data Structures	MA8351 - Discrete Mathematics	
3.	18.10.19 FRIDAY	CS8392 - Object Oriented Programming	EC8395 - Communication Engineering	CS8391 - Data Structures	CS8351 - Digital Principles and System Design
4.	19.10.19 SATURDAY	MA8351 - Discrete Mathematics		EC8395 - Communication Engineering	CS8392 - Object Oriented Programming
5.	25.10.19 FRIDAY	CS8391 - Data Structures	CS8351 - Digital Principles and System Design	CS8392 - Object Oriented Programming	EC8395 - Communication Engineering
6	26.10.19 SATURDAY	EC8395 - Communication Engineering	CS8392 - Object Oriented Programming	MA8351 - Discrete Mathematics	
7	28.10.19 MONDAY	CS8391 - Data Structures	CS8351 - Digital Principles and System Design	CS8392 - Object Oriented Programming	EC8395 - Communication Engineering

S.NO	DATE	SUBJECT CODE & NAME (FN)		SUBJECT CODE & NAME (AN)	
		9.15 a.m. to 12.30 p.m.		1. 15 p. m to 4.30p.m	
		Hall No.222	Hall No. 227	Hall No.222	Hall No. 227
8	29.10.19 TUESDAY	MA8351 - Discrete Mathematics		EC8395 - Communication Engineering	CS8392 - Object Oriented Programming
9	30.10.19 WEDNESDAY	CS8391 - Data Structures	CS8351 - Digital Principles and System Design	CS8392 - Object Oriented Programming	EC8395 - Communication Engineering
10	31.10.19 THURSDAY	EC8395 - Communication Engineering	CS8392 - Object Oriented Programming	MA8351 - Discrete Mathematics	
11	01.11.19 FRIDAY	CS8351 - Digital Principles and System Design	CS8391 - Data Structures	MA8351 - Discrete Mathematics	

SUB CODE	NAME OF THE SUBJECT	NAME OF THE STAFF	DEPT
MA8351	Discrete Mathematics	Ms.S.Geetha	MATHS
CS8351	Digital Principles and System Design	Mr.S.Rajarajan	CSE
CS8391	Data Structures	Ms.K.Abhirami	CSE
CS8392	Object Oriented Programming	Dr.D.Sivakumar	CSE
EC8395	Communication Engineering	Mr.T.Jeyaseelan	ECE


Class Coordinator


HOD/CSE



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
ACADEMIC YEAR 2019 - 2020 (ODD SEMESTER)
RETEST - I- ATTENDANCE SHEET

31/7/19

SUB CODE / SUBJECT: CS8391, Data Structures

YEAR / SEM: II / III

S.No.	Roll No	Student Name	AT Mark Out of 50	Retest Attendance	Retest Mark
1.	11	Deepika/V	22	/	39
2.	19	Jemima Esther Grace.M	19	/	38
3.	20	Kamala Priya.M	27	/	40
4.	21	Kaviyacehlvan.K	27	/	35
5.	23	Nandha Kishore.S	18	/	38
6.	24	Nasrin Banu.N	24	/	35
7.	26	Naveen Sundhar.V	7	/	21
8.	28	Priya.G	23	/	35
9.	29	Priyadharshini.M	24	/	37
10.	31	Ramya.I	19	/	35
11.	35	Sivaranjani.T	29	/	38
12.	26	Sneha.K	18	/	34
13.	38	Sowmiya.K	25	/	36
14.	39	Srinithi.C	27	/	36
15.	44	Gopinathan.D	AB	-	-
16.	45	Sindhu.R	15	/	27
No of Students Present				15	
No of Students Absent				1	
Staff Sign				Kodak 31/7	
HOD Sign				S S	



K. Cole
Staff Signature

phase-I
Batch-I - 5 slots
Batch-II - 4 slots
9 slots



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ACADEMIC YEAR 2019-20 / ODD SEMESTER
CS8391 - DATA STRUCTURES - REVISION TEST MARK STATEMENT

S.NO	REGISTER NO	NAME	SLOT 1 (24)		SLOT 2 (25)		SLOT 3 (32)		SLOT 4 (30)		SLOT 5 (40)	
			ATT.	MARK	ATT.	MARK	ATT.	MARK	ATT.	MARK	ATT.	MARK
1.	821118104001	Abarna M	/	13	/	20	OD		/	16	/	32
2.	821118104002	Abinaya N	/	20	/	16	/	30	AB		/	28+4
3.	821118104003	Abirami K	/		/	25	/	32	/	23	/	28
4.	821118104004	Abirami M	/	13	/	20	/	21	/	15	/	32
5.	821118104005	Abirami P	/	18	/	23	/	21	/	18	/	38
6.	821118104006	Abu vanayaki T	/	16	/	23	/	31	/	22	/	40
7.	821118104007	Aiswarya K	OD		/	20+5	/	24	/	22	/	40
8.	821118104008	Archana A	OD		/	10	/	22	OD		/	25
9.	821118104009	Arunkumar G	/	20	/	17	/	24	/	18	/	23
10.	821118104010	Babu M	/	14	/	13	/	26	/	19	/	20
11.	821118104011	Deepika V	/	10	/	15	/	23	/	13	/	28
12.	821118104012	Dharshini T R	OD		/	19	/	24	/	12	/	24
13.	821118104013	Divyabharathi G	/	20	/	25	/	24+8	/	19	/	37
14.	821118104014	Gayathri J	OD		/	20	/	21	/	20	/	37
15.	821118104015	Gayathri K	/	22	/	19	/	22	/	19	/	38
16.	821118104016	Gunaseelan B	/	18	/	16	/	24	/	21	/	37
17.	821118104017	Hariharan E	/	17	/	13	/	24	OD		OD	
18.	821118104018	Iswarya S	/	19	/	15	/	17	/	12	/	27
19.	821118104019	Jemima Esther Grace M	/	13	/	16	/	24	/	11	AB	
20.	821118104020	Kamalapriya M	/	22	/	17+8	/	17+8	/	8+22	/	10
21.	821118104021	Kaviyachelvan K	/	14	/	09	/	21	/	19	/	21
22.	821118104022	Keerthana V	/	20	/	16+8	/	31	/	25	/	40

OD-4
F-1

AB-101
F-2

A-101
F-2

A-1
O-2
F-6

A-1
OD-1

S.NO	REGISTER NO	NAME	SLOT 1 (24)		SLOT 2 (25)		SLOT 3 (32)		SLOT 4 (30)		SLOT 5 (40)	
			ATT.	MARK	ATT.	MARK	ATT.	MARK	ATT.	MARK	ATT.	MARK
23.	821118104024	Nandha kishore S	/	14	/	22✓	/	18	OD			
24.	821118104025	Narsrin banu N	OD	22✓	/	17	/	20	/	28✓		
25.	821118104026	Nathiya devi N	/	22✓	/	24✓	/	22	/	15		
(26.)	821118104027	Naveen sundhar V	/	13	/	13	/	11	/	4		
27.	821118104028	Pathmanaban R	/	16	/	24✓	/	22	/	22		
28.	821118104029	Priya G	/	09	/	23✓	/	21	/	21		
29.	821118104030	Priyadharshini M	/	22	/	24✓	/	15	/	28		
30.	821118104031	Priyadharshini R	/	17	/	✓18+7	/	24	/	24		
31.	821118104032	Ramya J	/	12	/	18	/	22	/	8		
32.	821118104033	Rathnakumar T	/	15	/	23✓	/	18	/	11		
33.	821118104034	Sarathkumar R	/	16	/	10	/	18	/	13		
34.	821118104035	Selvarani T	OD		/	25✓	/	24	/	28		
35.	821118104036	Sivaranjani T	/	12	/	22✓	/	16	/	9		
36.	821118104037	Sneha K	/	15	/	23	/	22	/	11		
37.	821118104038	Sneka S	/	12	/	24✓	/	21	/	14		
38.	821118104039	Sowmiya K	/	10	/	23✓	/	18	/	11		
39.	821118104040	Srinithi C	/	06	/	16	/	21	/	13		
40.	821118104041	Suruthi S	/	16	/	23✓	/	22	/	19		
41.	821118104042	Surya R	/	15	/	22✓	/	23	/	25		
42.	821118104043	Vaishnavi P	/	11	/	23✓	/	18	/	12		
43.	821118104044	Vithyatharan S	/	16	/	20	/	19	/	23		
(44.)		Gopinathan.D	/		/	08	1AB	.	/	11		
45.		Sindu.P	/	21✓	/	22✓	AB	11+14	/	24		
		Staff Sign	Kidde		Kidde		Kidde		Kidde	Kidde		
		HoD Sign		SS		SS		SS		SS		

Date(s)

18.9.19 (B1, B2)
FN
(Merged due to
IIFPT visit)

Unit - IV

OD - 2

F - 2

19.9.19 (AN) - B2
25.9.19 (FN) - B1
Unit - V

AB - -

F - 3

26.9.19 (FN) - B1
24.9.19 (AN) - B2
Unit - VI

AB - 1

F - 3

30.9.19 (AN) - B1
27.9.19 (AN) - B2
Unit - VII

AB - -

F - 10

OD - 1

1/10/19 (FN) - B1
Unit - I, II

CS8391- Data Structures
Unit-I

1. procedure to perform operations on a list
 - a. using array
 - b. using linked list.
2. procedure to perform operations on
 - a. circular linked list.
 - b. Doubly linked list.
3. polynomial manipulation
4. a. Merging of 2 list
b. Reversing a list
5. Student details / Employee details using list.

Slot 6

Unit-II

1. Array representation & operation on stack
2. linked list representation & operation on stack
3. stack Application
 1. parenthesis checker
 2. Conversion of expression
 3. Evaluation of expression
4. Array representation, linked list representation of Queue
5. Circular Queue
priority Queue
Deque } Implementation & Operation

Slot 5

Unit-III

1. Binary tree, representation
2. Tree Traversal **
3. Binary search tree & its operations ** Slot-3,4
4. Threaded tree & Expression trees.
5. AVL trees & balancing **
6. B tree & B+ tree operations **
7. Heap Construction **

Unit-IV

1. BFS & DFS traversal **
2. Topological sorting ** Slot-2
3. Euler's circuit. **
4. shortest path algorithms X.X
 1. MST
 2. Prim's algorithm
 3. Kruskal's algorithm
 4. Dijkstra's algorithm
5. Graph Representation.

Unit-V

1. Searching - Binary Search, Linear Search Slot-2
2. Sorting - Bubble Sort, Selection Sort, Insertion Sort
Radix Sort, Shell Sort.
3. Hashing - Functions.
4. Separate chaining, Open Addressing
5. Rehashing, Extensible hashing,

Name : G. Anandkumar
Std : CSE - 1st year
Sub : Data structure [Test Note]
Roll No. : 09

**SHALU
INDEX**

① Insertions:- Array Implementation

(i) Insertion at the beginning of an array:-

begin

If $N = \text{MAX}$, return

ELSE

$N = N + 1$

for all elements from A

move to next adjacent location

$A[\text{FIRST}] = \text{New_element}$

end

(ii) Insertion at the Given index of an array:-

begin

If $N = \text{MAX}$, return

ELSE

$N = N + 1$

SEEK location index

for all elements from $A[\text{index}]$ to $A[N]$

move to next adjacent location

$A[\text{index}] = \text{New_element}$

end

(iii) Insertion at After the given index of an array:-

begin

If $N = \text{MAX}$, return

ELSE

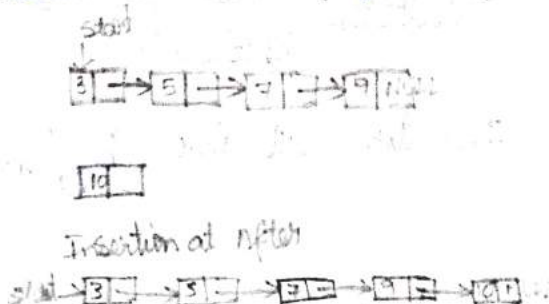
$N = N + 1$

SEEK location index

FOR all elements from $A[\text{index} + 1]$ to $A[N]$
move to next adjacent location.

$A[\text{index} + 1] = \text{New_element}$

end.



(iv) Insertion at before the given index of an array:-

begin

If $N = \text{MAX}$, return

ELSE

$N = N + 1$

SEEK location index

FOR all elements from $A[\text{index} - 1]$ to $A[N]$
move to next adjacent location.

$A[\text{index} - 1] = \text{New_element}$

end

continue legibly

(v) Alternate Method:-

Step 1 : start

Step 2 : set $J = N$

Step 3 : set $N = N + 1$

Step 4 : Repeat steps 5 & 6 while $J > K$

Step 5 : set $LA[J] = LA[J+1]$

Step 6 : set $J = J + 1$

Step 7 : set $LA[K] = ITEM$

Step 8 : stop.

(2) Deletion operation:-

Step 1 : start

Step 2 : set $J = K$

Step 3 : Repeat steps 4 & 5 while $J \leq N$

Step 4 : set $LA[J] = LA[J+1]$

Step 5 : set $J = J + 1$

Step 6 : $N = N - 1$

step 7: stop.

③ Search operation:

step 1: start

step 2: set $J = 0$

step 3: Repeat steps 4 & 5 while $J < N$

step 4: $LA[J]$ is equal ITEM THEN GOTO STEP 6.

step 5: $J = J + 1$

step 6: PRINT $J, ITEM.$

step 7: stop.

good.

12/12

K. K. K.

Part - A1. Significance of Data structure :-

- * Data structure is important because ^{it is used} of almost every programs or software system.
- * It helps to write a codes, structures the code and solve problems.
- * Data structure can be maintained easily by ~~encouraging~~ ^{better} the design or implementation.
- * Data structure is a container of data to store, manipulate and arrange.
- * It can be processed by algorithms.

3. Primitive data type:-

* Primitive data type is a data type involving almost every programming language.

* It is a basic data type in almost

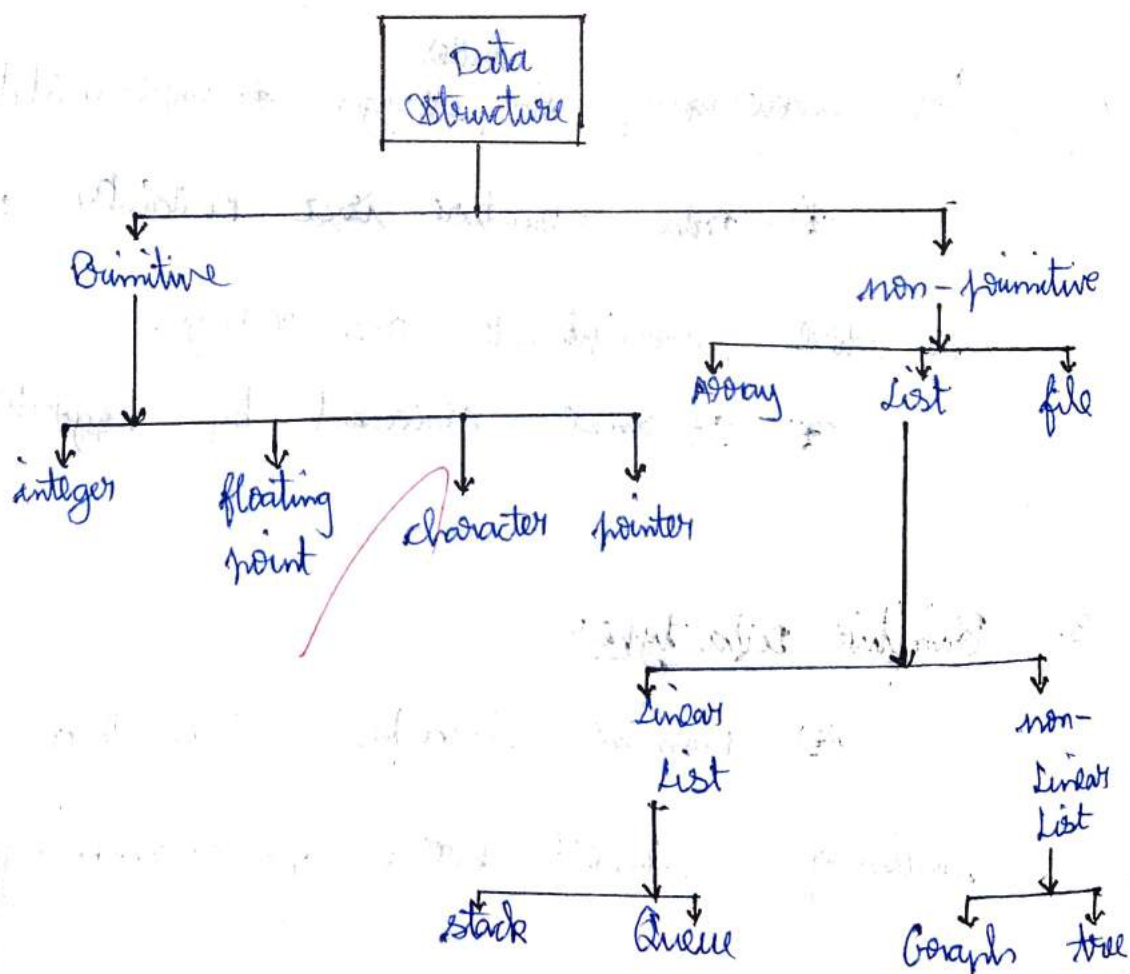
available in programming language.

(*) It is used to ~~return~~ represent a single value.

Example:-

integer, float, character, Boolean.

2. classification of Data Structure:-



Part - B

2. (i) To find the duplicate element in the linked list:-

* To count the elements, in a list of nodes are to be traversed. In between increment the counter by 1. If we reach NULL, all the nodes must be traversed. And the counter value can be displayed.

Algorithm:-

Step 1: [Initialise] set count = 0

Step 2: [Initialise] set PTR = START

Step 3: Repeat steps 4 and 5 while PTR != NULL

Step 4:

set count = count + 1

Step 5:

set PTR = PTR → next

[End of Loop]

Step 6: Write count

Step 7: Exit.

Step 12: Exit.

2.

(ii) Deletion:

Step 1: If START = NULL

write underflow

Step 2: go to step 5

End If

Step 2: SET PTR = START

Step 3: SET START = START → NULL

Step 4: Free PTR

END OF LOOP

Step 5: EXIT

(i) Count number of elements in the list :- [To find the duplicate element in the linked list]

Algorithm:-

Step 1: [INITIALIZE] SET COUNT = 0

Step 2: SET PTR = START

Step 3: Repeat steps 4 and 5 while PTR ≠ NULL

Step 4: SET COUNT = COUNT + 1

Step 5: SET PTR = PTR → NEXT

[END OF LOOP]

Step 6: Write COUNT

Step 7: EXIT.

05/09/2019

Revision Test

1. B tree - Insertion, Deletion, search operations :-

B tree :-

* A B tree is a specialized m -way tree it is widely used in the disk access.

* Storing the large number of keys in a single node and the height is small.

In addition the following properties :-

* Every node in the B tree have maximum (m) children.

* Every node in the B tree except root node and the leaf node have minimum (m) children.

* The root node have two children which is not a terminal leaf node.

* The leaf node is same in that level.

B tree operations:

(i)

searching of a element in B tree:-

* Searching of a B tree is similar to the Binary search tree. In searching a B tree the following steps are to be followed.

* Searching the element in B tree is '59'.

* The search value is 59. begin with the

root node. if the root node value is less than the search value. go to the right sub tree.

* The search value is 59, and the root node is 45. so go to right sub tree.

* In right sub tree they have two nodes 46 and 63.

* In the right sub tree 46, that is the left sub tree 63 in between we have tree nodes 57, 59 and 61. The search value

Repeat steps ④ & ⑤ while $pos > 1$

Step 4: Set $Heap[pos] = pos/2$

Step 5: If $Heap[pos] \leq Heap[pos]$

Go to step 6, otherwise

Else

Swap $Heap[pos]$, $Heap[pos]$

$pos = pos$

[End of If]

[End of loop]

Step 6: Return

(ii) Deletion on Binary Heap

Algorithm:

Step 1:

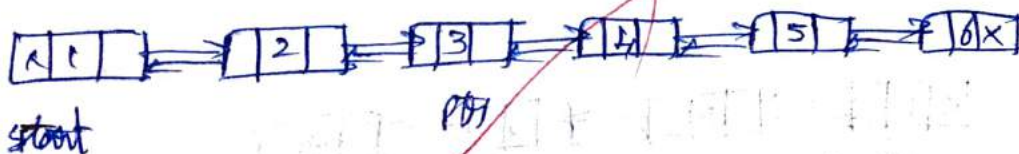
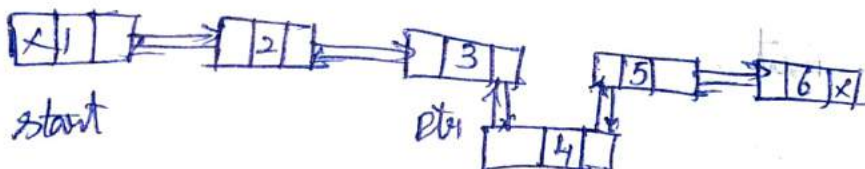
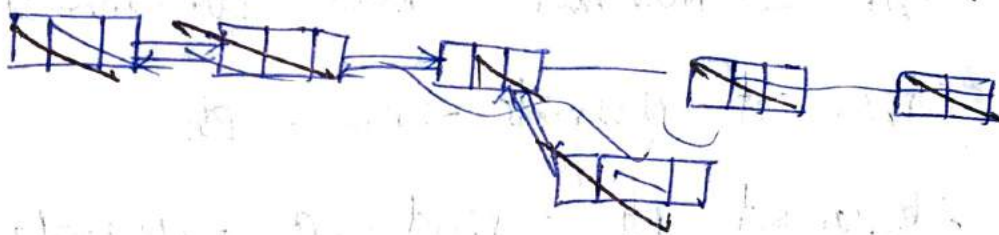
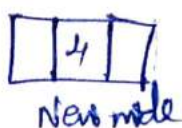
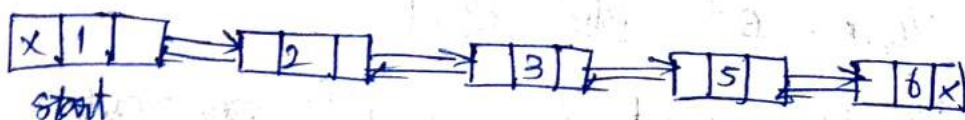


Kid

Model Set 2 Test
Answers

1.

a) Inserting an element after the given index in
Doubly Linked List.



Algorithm:

Step 1: If avail = Null
write overflow
Go to step II

[End of I]

Step 2: set new node $\rightarrow$ avail

Step 3: set avail $\rightarrow$ next

Step 4: set new node $\rightarrow$ Data = val

Step 5: set ptr = start

Step 6: Repeat steps 7 while ptr $\rightarrow$ Data $\neq$ Num

Step 7: set ptr $\rightarrow$ ptr $\rightarrow$ next

Step 8: set new node $\rightarrow$ next = ptr $\rightarrow$ next

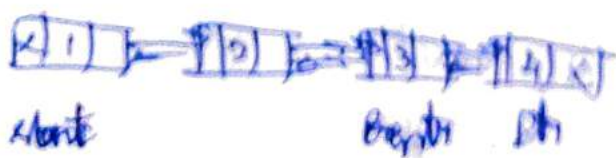
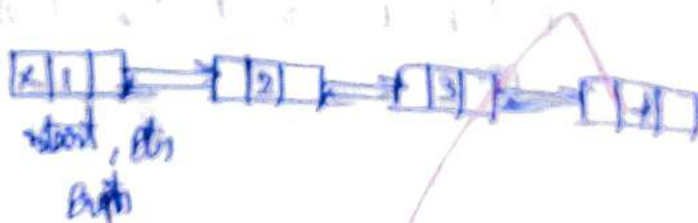
Step 9: set new node $\rightarrow$ prev = ptr

Step 10: set ptr $\rightarrow$ next $\rightarrow$ prev = new node

Step 11: Exit -

(b)

Deleting a last node in doubly linked list :-



Algorithm:

Step 1: If start = Null
write underflow
Go to 7
[End of 1]

Step 2: Set start = ptr

Step 3: Repeat Step 4 while ptr → next! = Null

Step 4: Set ptr = ptr → next

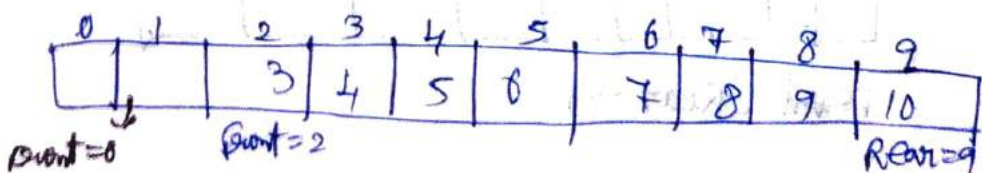
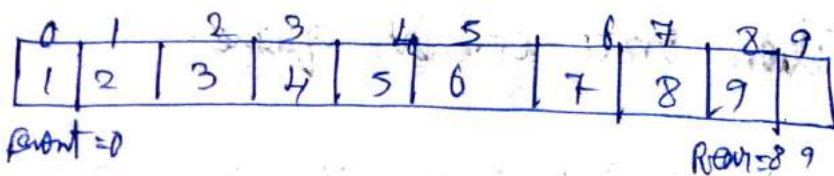
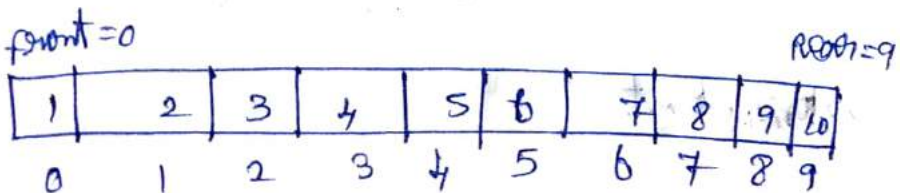
Step 5: Set Next = Null

Step 6: Free ptr

Step 7: Exit

2.

Insertion into Circular Queue:-



• R rotations have three types

R-1

R0

R1

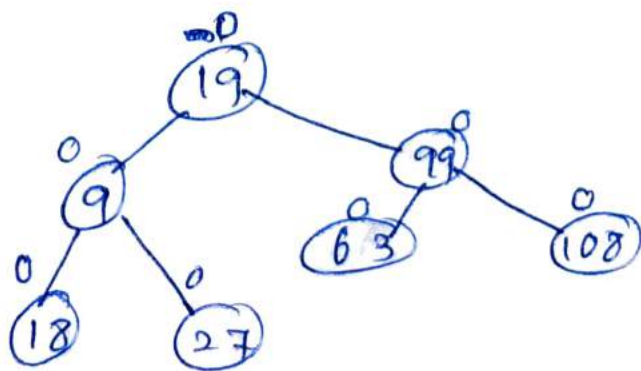
(*) In L and R rotations, deleting deletions are perform like those above types of L and R rotations.

Deletion on AVL Tree

use step 10, delete nodes

47
60

Kedde
19/10



NAME G. Divyabharathi Std. IInd year

Roll No. 13 Sec. DS

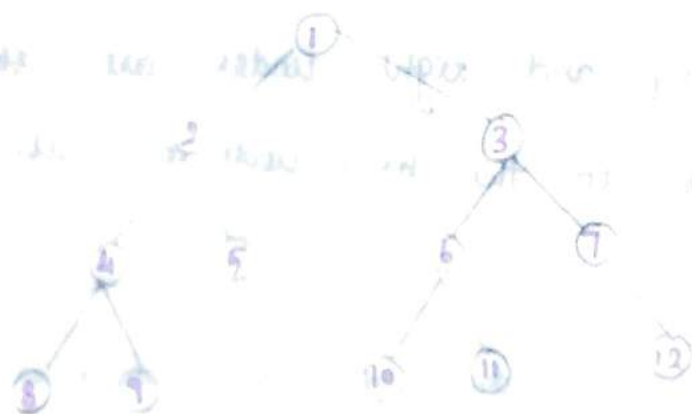
1.

Bounded Test

Threaded trees are same as binary

Search trees

The difference in binary and threaded is
in threaded the null pointers is stored.



Example tree for binary tree

The tree can also be represented in linked list
now we see the binary tree represented in
the linked list formation



without threaded tree
linked list representation

In threaded binary tree two types are there.

One is One-way threaded tree
or

Singe threaded tree.

Another one is two-way threaded tree
or

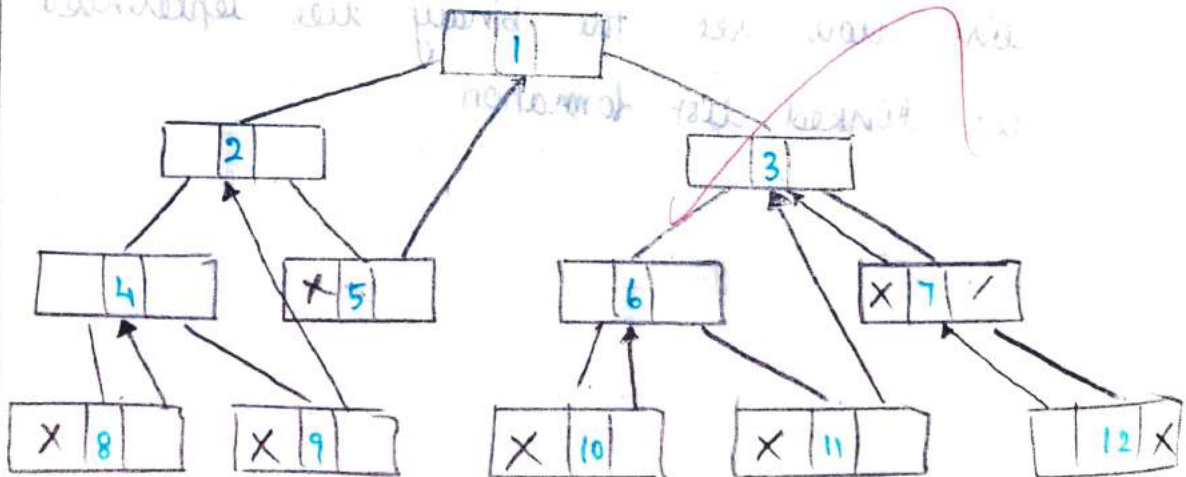
Double threaded tree.

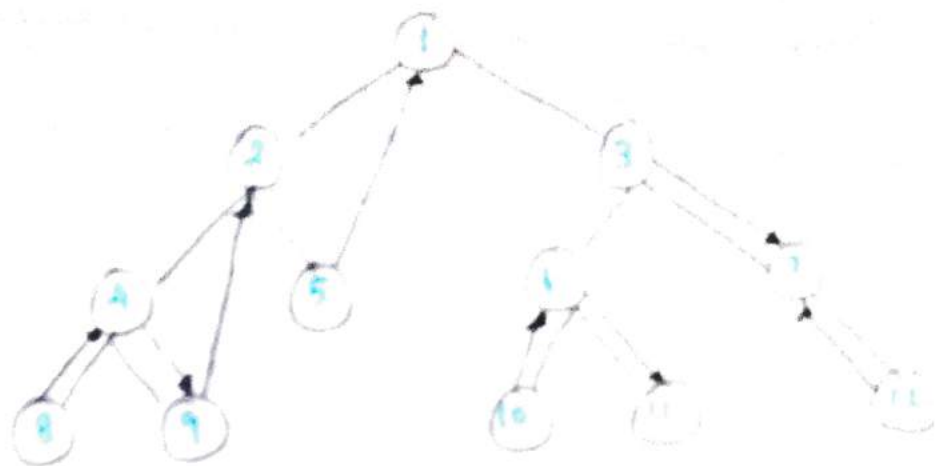
From the above tree linked list representation the left and right nodes are store the pointers to the next node or its child node.

One way threaded or Singe threaded node.

Inorder traversal for the above tree representation

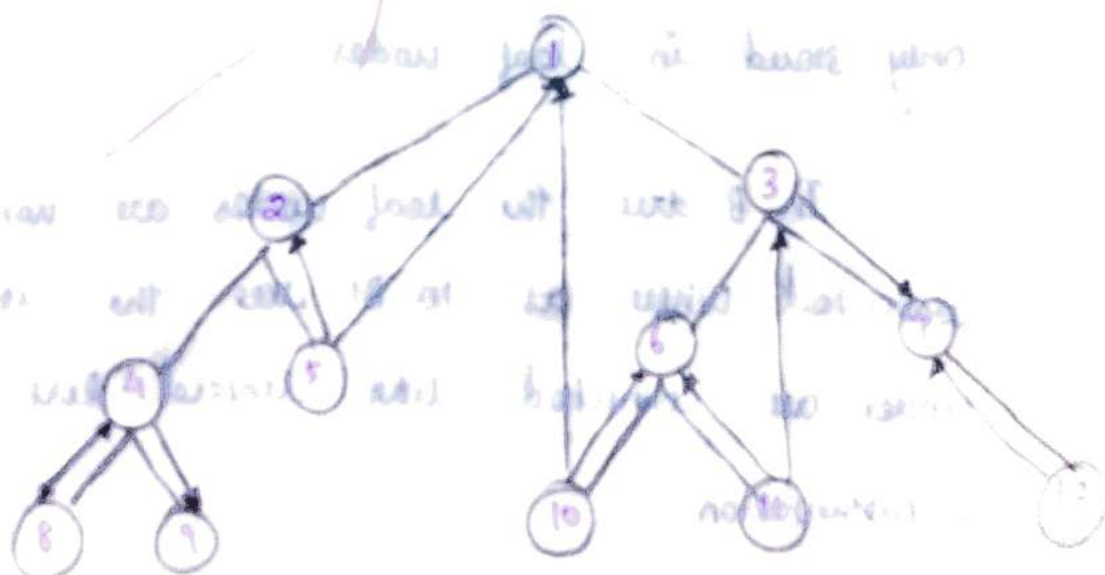
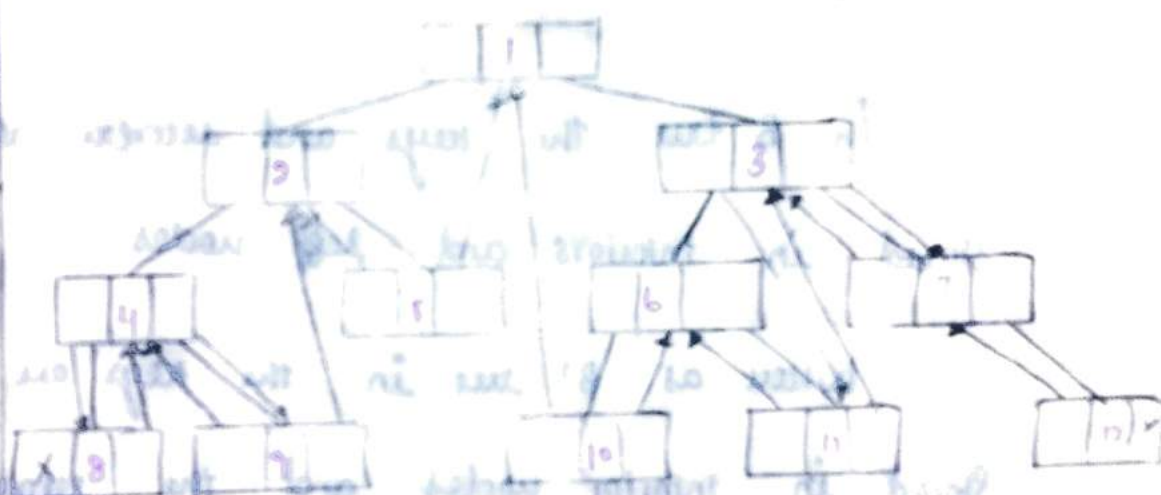
8, 4, 9, 2, 5, 1, 10, 6, 11, 3, 7, 12

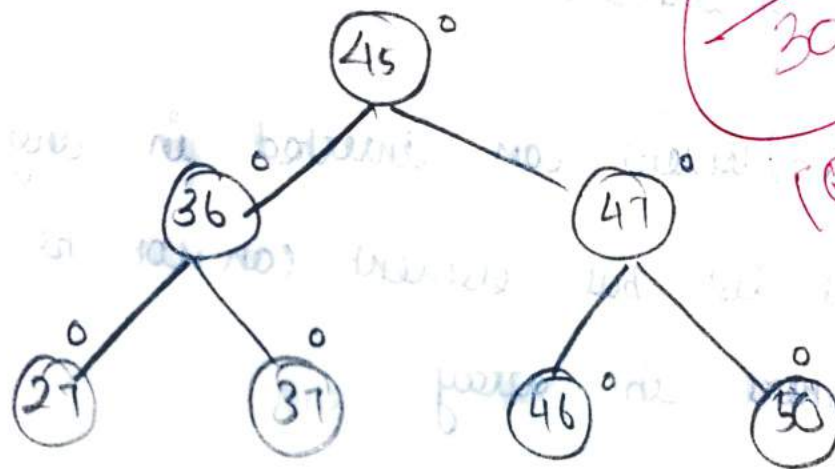
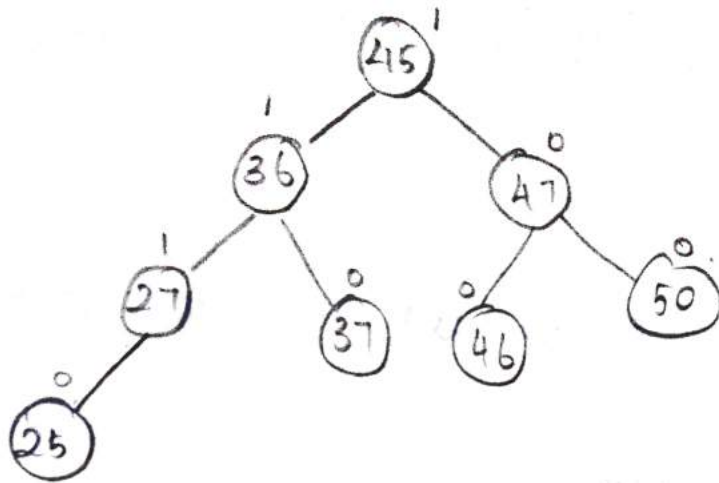




In tree representation

Double way threaded or Two way threaded
binomial





$$\frac{19}{30}$$

10-pending
K. ylle
1/10

Balanced AVL tree.

2

If the insertion is violated the
AVL trees then we have to
provide 4 moves to avoid the
violation in the insertion process.

1) Insertion in left subtree in left child

1. Circular Linked List

~ Circular linked list is a linked list representation.

~ Elements can be inserted anywhere in linked list but element cannot be inserted anywhere in array.

~ Because array is a contiguous memory.

~ Linked list is dynamic memory allocation. we can allocate memory any time we required.

~ In Circular linked list a previous node stores the address of the next node.

~ And the last node stores the address of the starting node.

~ It is made like a chain, this is known as circular linked list.

Applications of Circular linked list

✓ Circular linked list used in one of the main in our personal computers. In our personal computer we have multiple application. These multiple applications are running. If memory fixed it is having the structure of linked list circular.

✓ Another one application is multiplayer game.

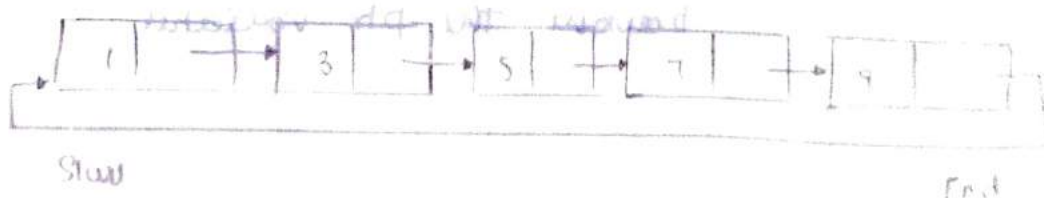
✓ Also circular linked list used to create circular queue also.

Insertion:

Insertion at the beginning of the circular linked list.

Inserting a node at the end of the circular linked list.

Circular linked list structure



3.

Azithmetic Expression

Conversion and Evaluation:

Algorithm for infix to postfix conversion:

Step 1: Add '(' at the end of postfix expression

Step 2: push '(' at the starting in the stack

Step 3: Repeat steps until each character in infix expression is scanned.

Step 4: If operand is encountered push into the stack. postfix expression

Step 5: if ')' is encountered,

a. repeat pop from the element from the stack and push into the stack until the '(' is encountered in the stack.

b. The '(' is removed it does not push into the stack.

Step 6: If an operator o is encountered

(a): pop the element from the stack and add to the postfix expression if it is higher precedence.

Step 3: The same process in
infix to postfix is moved in
the step 3.

(A+B)/C*D

This is the procedure of infix to
prefix expression.

37
40

KCB

18.10.2019

MODEL Q. TEST

1. Doubly linked list.

- (1) insertion of an element after given position
- (2) deletion of an element in the last position
- (3) searching.

2. Advantages of Circular queue over linear queue.

1) insertion and deletion procedure.

3. Diff. b/w B tree & B+ tree.

3, 14, 7, 1, 8, 5, 11, 17, 13, 6, 23, 12, 20, 26, 4, 16, 18,
24, 25, and 19

1. Doubly linked list :

Linked list having two parts. One is data and another one is the next node address part.

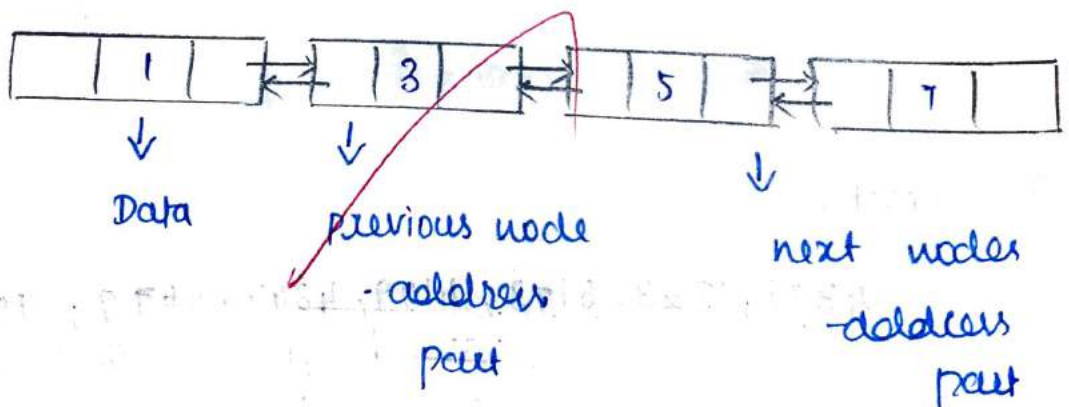
Doubly linked list having three parts in a node.

One is data part.

Second one is the next node address part.

And the third one is the previous node address part.

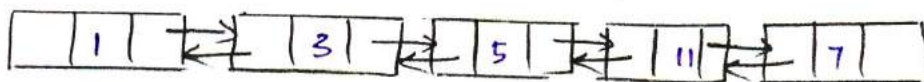
Example structure of doubly linked list



Insertion:

Insertion of an element in after the given node in doubly linked list.

Step 5:



Algorithm:

Step 1: if start = Null

wait for underflow

[end of if] go to step 6.

Step 2: set start = ptr

Step 3: Repeat steps while ptr != Null

set ptr = ptr -> next

Step 4: set ptr -> previous = Null

Step 5: Free ptr

Step 6: Exit

(iii) Searching operation:

Step 1: set start = ptr

Step 2: set Val = Num

Step 3: Repeat steps while ptr != Null

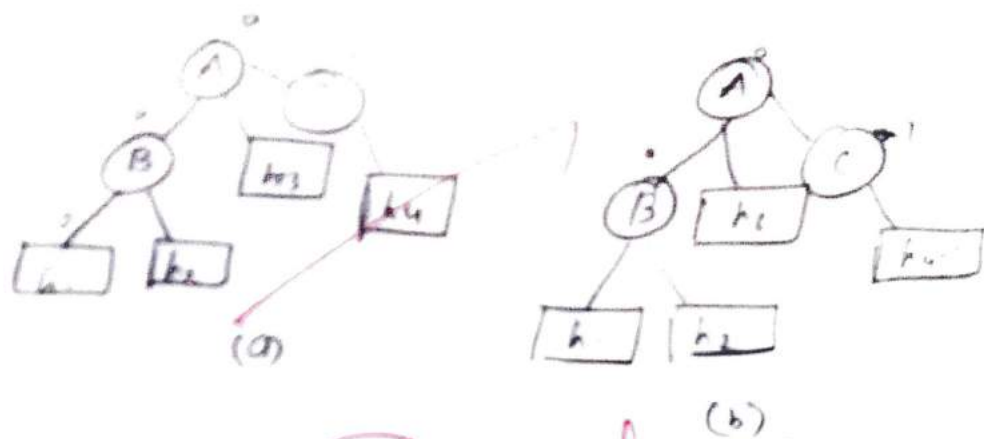
if ptr = ptr -> next

else

pos = ptr

Step 4: exit.

Ans: rotation

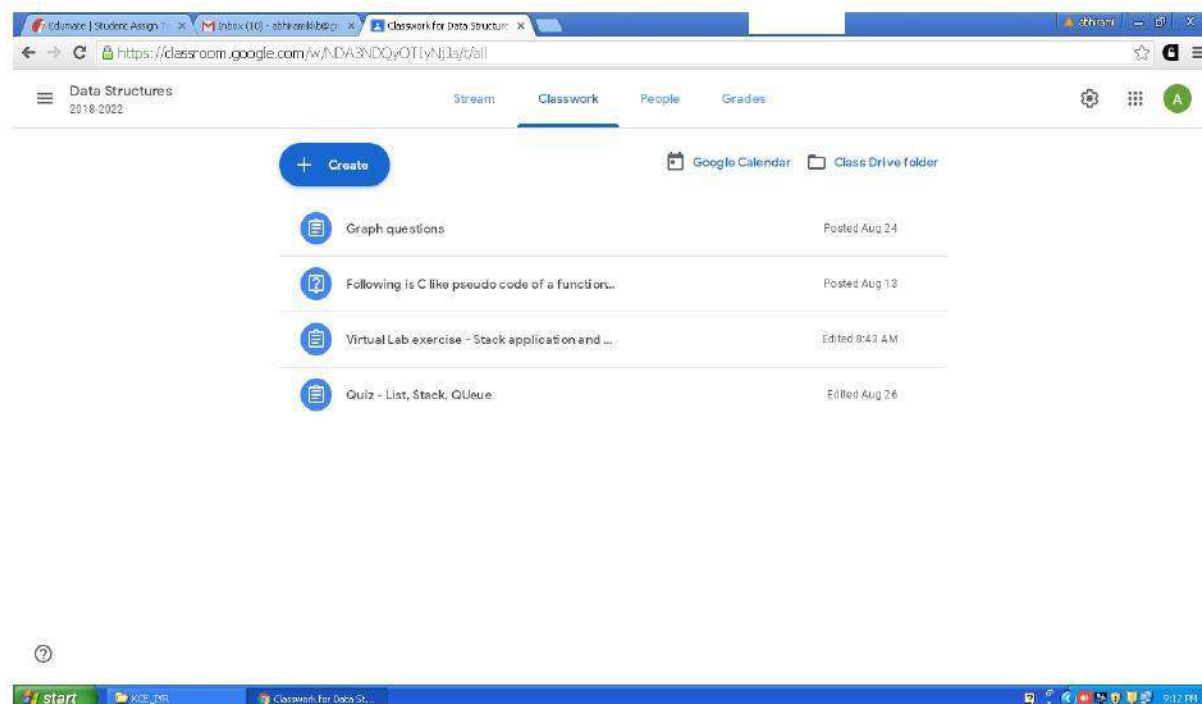


57
60

V. good
K. S. D.
19/10

PCE SKILL ACTIVITY EXECUTION –GOOGLE CLASSROOM BASED

(17.8.19 & 24.8.19)



PCE SKILL ACTIVITY EXECUTION SUMMARY

ACTIVITY	MODE	OUTCOME	MARKS ALLOTTED	NO. OF STUDENTS APPEARED
Application of Graph Concept No. of Questions: 8	Written (Advanced Learners)	Ability to apply graph concepts to solve real world problem	5	09
GATE Questionnaire - quiz No. of Questions: 10	Online	Ability to gain indepth knowledge and become competent for GATE exams	5	45*
Virtual Lab exercise No. of question:2	Online	Ability to convert arithmetic expressions into different notations Gain skills to apply Stack data structure and solve problems	10	45*
Coding Exercise No. of questions:1	Online	Ability to trace programming exercise, use data structure and find output	5	45*
Total Marks (appeared by all students)			20	-



KINGS
COLLEGE OF ENGINEERING
(NAAC Accredited Institution)
(Approved by AICTE, New Delhi, Affiliated to Anna University, Chennai)



ASSESSMENT TEST-I / II / MODEL EXAMINATION

**REGISTER
NUMBER**

8 2 1 1 1 8 1 0 4 0 3 1

ROLL NO.

30

YEAR / BRANCH / SECTION

II CSE

College Code & Name	8 2 1 1	Kings college of Engineering	
Degree/Branch	BE - CSE		
Subject Code	CS8391	Subject Title	Data Structures

Semester	III
Date & session	24/7/19 FN
No. of pages used	11

All the particulars given are verified	
Signature of the Invigilator with date	[Signature] 24/7/19
Name of the Invigilator	S. KARTHI Ap. mech

Instructions to the candidates

1. You are prohibited from writing your **NAME** in any part of the answer book.
2. You are prohibited from writing or leaving any distinguishing marks so as to identify your answer book.
3. Use both side of the paper for answering questions (Except front page).
4. Check the regulation, Degree, Branch, Semester, Subject code and Subject Title of the Question Paper before answering the questions.
5. Possession of any incriminating material and Malpractice of any nature shall be punishable as rules.
6. **No additional sheets will be provided.**

Signature of the Student with Date after Evaluation

R. Pragyadharshini / 26/7/19

SPACE FOR MARKS

48

50

100

Marks in words for 50 / 100

Signature of the Examiner with Date

K. ABHIRAM 26/7/19

Name

K. ABHIRAM

Excellent. Keep it up.
Wish for O grade in Uty
exams.

Part-A

1.	linear array	linked list.
	Array is a collection of same data type of element	linked list is a collect of data item which is connected by list
	It is accessed randomly	It is accessed sequentially not randomly.
	It has single dimensional, multidimensional and two dimensional array	It has singly, doubly, circularly linked list
2.	Memory allocates at compile time	Memory allocates at run time.
we will used to prefer linked list. because it is a collection of all data item. It is ordered manner. It allocates pointer and extra memory storage.		

2.	Feature	Linear	Non linear.
	Definition.	Linear data structure is defined as arranged in a sequential order	Non-linear data structure are arranged in the sorted order
	Traversing of the data	Traversing of the data at a one time (single run)	Traversing of the data at one go is not possible

Ease of implementation	Simple	complex
Levels involved	Single	multiple
Example.	Stack, queue, list, etc	tree and graph

4. priority Queues.

* priority Queues is a extension of Queues are follows.

* priority Queues are arranged in a order

* If a high priority element is arranged dequeued before the low priority.

* If the elements are same in priority then they are accessed sequentially.

* priority Queues can be implemented

- using array
- using heaps

(3) Doubly linked list more useful than a Singly linked list

Doubly linked list more useful than a Singly linked list because it enables the item forward and backward direction to access. It is a more complex and expensive operation. It has two linked list.

now dup

Step 4: set $ptr2 \rightarrow next = ptr2 \rightarrow next$

Step 5: set $ptr1 = ptr1 \rightarrow next$

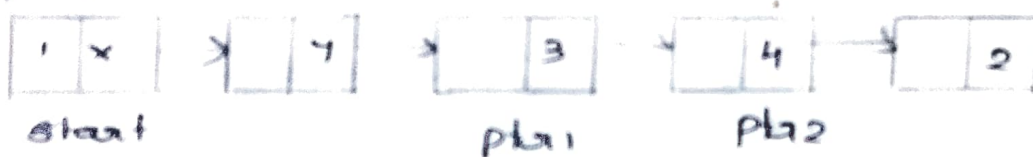
Illustration:



Start

$ptr1$

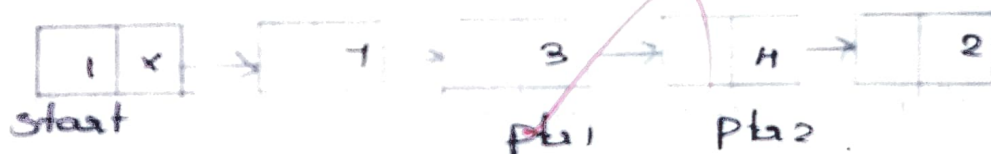
$ptr2$



Start



Duplicate



(ii)

Swap 2 adjacent

Step 1: start

Step 2: before set After $ptr \rightarrow next =$ before $ptr \rightarrow next$

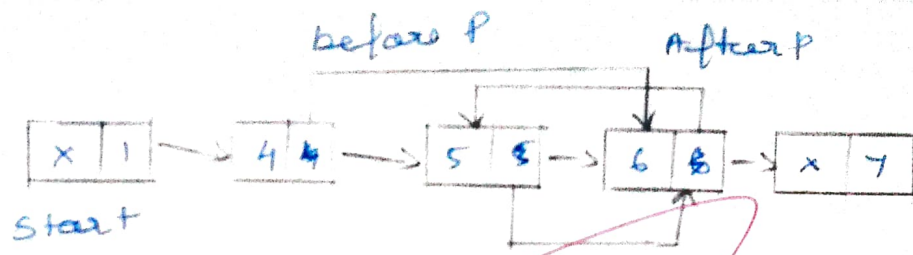
Step 3: set After $ptr \rightarrow next = ptr$

Step 3: Set After $ptr \rightarrow prev = P$

Step 4: set $prev \rightarrow next =$ After swap

Step 5: set before $ptr \rightarrow next =$ after $ptr \rightarrow next$

Step 6: End





ASSESSMENT TEST-I / II / MODEL EXAMINATION

REGISTER
NUMBER

8 2 1 1 1 8 1 0 4 0 4 3

ROLL NO.

42

YEAR / BRANCH / SECTION

II / CSE

College Code & Name	8 2 1 1	kings college of Engineering
Degree/Branch	B.E - Computer Science Engineering	
Subject Code	CS8391	Subject Title Data Structure

Semester	III
Date & session	24/7/19 - FN
No. of pages used	88 12

All the particulars given are verified	
Signature of the Invigilator with date	<i>[Signature]</i> 24/7/19
Name of the Invigilator	K. ABHIRAM

Instructions to the candidates

1. You are prohibited from writing your **NAME** in any part of the answer book.
2. You are prohibited from writing or leaving any distinguishing marks so as to identify your answer book.
3. Use both side of the paper for answering questions (Except front page).
4. Check the regulation, Degree, Branch, Semester, Subject code and Subject Title of the Question Paper before answering the questions.
5. Possession of any incriminating material and Malpractice of any nature shall be punishable as rules.
6. No additional sheets will be provided.

Signature of the Student with Date after Evaluation

P. Vaishnavi
26/7/19

SPACE FOR MARKS

30

50

100

Marks in words for 50 / 100

Signature of the Examiner with Date

[Signature] 26/7/19

Name

K. ABHIRAM

can do better - Aim for better performance

Part - A

1)

Array	linked list
* Array is a collection of similar data types.	* linked list is a ordered form with linked.
* Access to the elements of randomly.	* Access to the elements of sequentially.
* <u>Operator does not used, Here not require space.</u>	* Operator was used, Here require extra space.
* If the size of elements is fixed in the array.	* If the elements of the list was traverse.

2)

linear

non-linear

* Traversing of the elements to access the single time.

* Traversing of the elements to access the multi time.

* levels involved single.

* levels involved multi.

⇒ Examples:

* array, stack, Queue, linked list.

⇒ Examples:

* tree and graph

3)

Doubly linked list more useful. because,

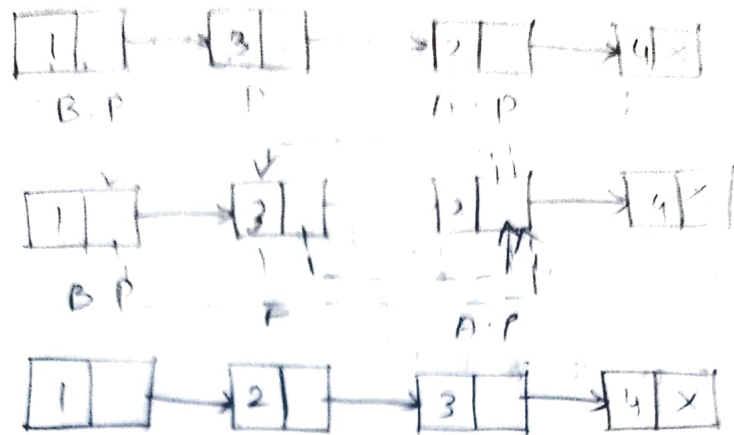
* Traversing the element in linked list both sides forward and backward.

* We want insert the newnode at the beginning, it was easily.

* Deleting operation is more efficient.

8/ (a)

(ii) Swap :



Procedure :

{

Void

Swap with position

Position P, After P

~~P = next~~

P = Before P → next

After P = P → next

P' → next = After P → next

After P' → next = P

~~P' = After P → next~~

~~After P' = Before P → next~~

Before P → next = After P

} End



ASSESSMENT TEST-I / II / MODEL EXAMINATION

**REGISTER
NUMBER**

8 2 1 1 1 8 1 0 4 0 3 5

ROLL NO.	18AD34
YEAR / BRANCH / SECTION	II / CSE / A

College Code & Name	8 2 1 1 Kings College of Engineering
Degree/Branch	B-E / CSE
Subject Code	CS8391
Subject Title	Data Structures

Semester	III
Date & session	6-9-19 / FN
No. of pages used	

All the particulars given are verified	
Signature of the Invigilator with date	<i>[Signature]</i> 6/9/19
Name of the Invigilator	D. VENNILA

Instructions to the candidates

1. You are prohibited from writing your **NAME** in any part of the answer book.
2. You are prohibited from writing or leaving any distinguishing marks so as to identify your answer book.
3. Use both side of the paper for answering questions (Except front page).
4. Check the regulation, Degree, Branch, Semester, Subject code and Subject Title of the Question Paper before answering the questions.
5. Possession of any incriminating material and Malpractice of any nature shall be punishable as rules.
6. No additional sheets will be provided.

<u>Signature of the Student with Date after Evaluation</u> T. Selvarani 9/9/19

SPACE FOR MARKS	
50	100
<u>Marks in words for 50 + 100</u> SEVEN ZERO	
<u>Signature of the Examiner with Date</u> <i>[Signature]</i> 9/9/19	
Name	K. ABHIRAM

sim for better score in Uty. exam.

Part-B

11.

a. (1) Circular queue:-

The circular queue is a queue in which the rear is adjacent to front position.

It reduces the emptiness of a queue.

Operations on circular queue:

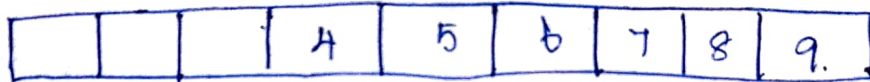
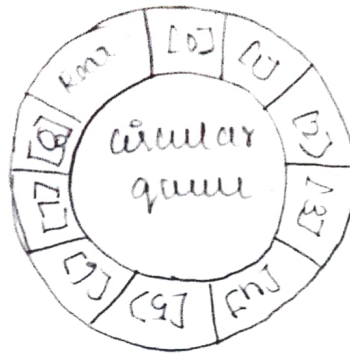
There are some operations for circular queue. They are

⇒ Insertion

⇒ Deletion

Insertion operation:

In circular queue, the insertion process is processed by checking the completely filled or partially filled else $\text{front} = 0$ and $\text{Rear} = \text{max} - 1$



Front

Front=3

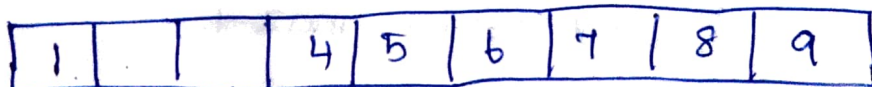
Rear

In this, if we processed to enqueue an element in the queue, it will show the queue is overflow.

To eradicate this situation, the elements in the queue are left shifted but it takes a long time in case queue is quite large.

This situation is overcome by inserting an element at the Rear=0.

val = 1



Rear=0

Algorithm:-

Step 1: If Queue = Null

Set Front = 0 and Rear = 0.

Set Val = Front = Rear.

END IF.

Step 2: If Rear = Max - 1 and Front $\neq$ 0

Set Rear = 0.

Set Rear[0] = Val.

Step 3: ELSE

Set Front = Front + 1.

Step 4: Exit.

Deletion Operation:

In deletion operation of queue, the elements are dequeued at the front position of a queue.

72	76	77	78	79	80	82	83	81
----	----	----	----	----	----	----	----	----

Front = 0

Rear = Max - 1

Part-C .

(01)

16.b.

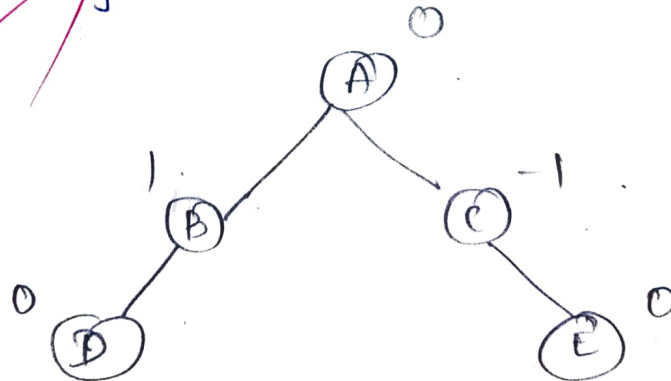
(9)

(ii) AVL trees:-

The AVL trees are specialised trees of taking the states like $-1, 0, 1$.

The Adelson, Velsky and Landis in 1972 discovered so it is called as AVL tree.

AVL tree is also called as balancing tree.





Good efforts - Aim for better performance in Ufy exams.

Part-8

11.6 (i) Array :

Array is a collection of similar data type elements.

Various operations of array :

* Insertion.

* Deletion

* Searching.

* Updation.

Insert an element in the middle of the array:

Step 1 : Start.

Step 2 : If $N = \text{MAX}$
Go to step 7.

Step 3 : Repeat step 4 ~~for~~ from $A[\text{index}]$ to N .

Step 4 : Swap all the elements from
 $A[\text{index}]$ to N to the next
adjacent location.

Step 5 : Set $N = N + 1$.

Step 6 : Set $A[\text{index}] = \text{val}$

Step 7 : Stop.

Step 1

0	1	2	3	4
23	45	36	54	

insert 40 at index 2

Step 2

0	1	2	3	4
23	45		36	54

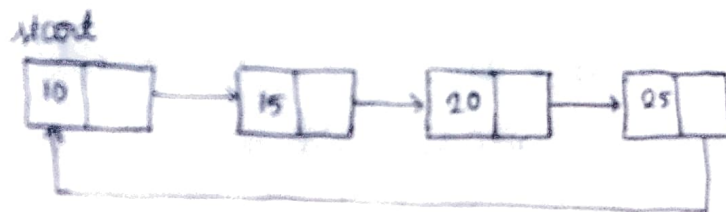
Step 3

$A[2] = 40$

0	1	2	3	4
23	45	40	36	54

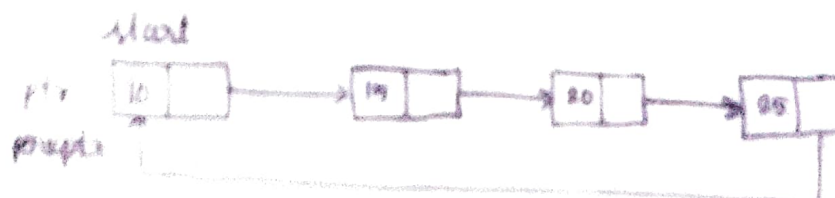
ii) deleting last node :

Step 1



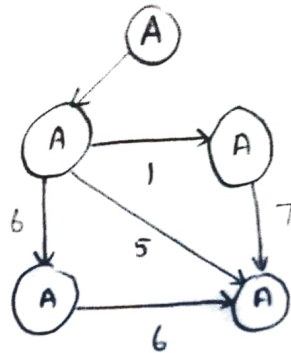
delete 25

Step 2



14. a. (i) Dijkstra's algorithm:

The given graph is directed.



* Initially set order of $A = 0$ and $N = \{A\}$

* If the paths are $A = 1$, $A = 5$, $A = 6$

$N = \{A, A(1)\}$

* If the paths are $A = 5$, $A = 6$, $A = 12$, $A = 8$

$N = \{A, A(1), A(5)\}$

* If the paths are $A = 6$, $A = 12$, $A = 8$

$N = \{A, A(1), A(5), A(6)\}$

* If the paths are $A = 12$, $A = 8$

$N = \{A, A(1), A(5), A(6), A(8)\}$

$N = \{A, A(1), A(5), A(6), A(8), A(12)\}$

Algorithm:

Step 1: Set the order of starting node is 0.

Step 2: Put it onto N.

Step 3: Find all the paths.

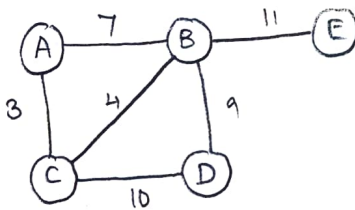
Step 4: Select the minimum beyond the weight of the edge.

Step 5: Add it to the N.

Step 6: Repeat this process until the destination is obtained.

Step 7: Stop.

ii)



$$F = \{A, B, C, D, E\}$$

$$T = \{(A, C), (B, C), (A, B), (B, D), (C, D), (B, E)\}$$

Procedure:

Step 1: Construct a forest using the vertices and construct tree ^{stack} using edges.

Root - A

3. Operations on Queue:

- * Enqueue.
- * Dequeue
- * Search.

4. Postfix form:

3 3 4 1 - / * 6 2 * +

5. height of the tree:

height of the tree = height left sub tree + height of right sub tree + 1

depth of the node:

The number of nodes between the root node and designation node.

degree of node:

The number of edges connected to the node is called degree of node.

edge:

A path between two nodes is called edge.

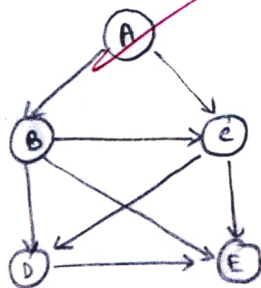
6. Binary tree :

- * Binary tree is a well order tree.
- * The value of left sub tree is less than root node.
- * The value of right sub tree is greater than root node.
- * Every node except the leaf node have atmost two children
- * Leaf nodes are in the same level

7. Adjacency list of a graph :

Adjacency means there exists a path, to reach the node.

Ex :



Adjacency list :

A : B, C

B : C, E, D



Reg. No. :

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Question Paper Code : 90152

B.E./B.Tech. DEGREE EXAMINATIONS, NOVEMBER/DECEMBER 2019

Third Semester

Computer Science and Engineering

CS8391 – DATA STRUCTURES

(Common to : Computer and Communication Engineering/Information Technology)

(Regulations 2017)

Time : Three Hours

Maximum : 100 Marks

Answer ALL questions

PART – A

(10×2=20 Marks)

1. Define Linked List.
2. Define an Abstract Data Type.
3. List the applications of stacks.
4. State the rules to be followed during infix to postfix conversions.
5. What do you mean by level of the tree ?
6. Define a binary search tree.
7. What is meant by strongly connected in a graph ?
8. Define adjacency list.
9. What do you mean by internal and external sorting ?
10. Define radix sort.

PART – B

(5×13=65 Marks)

11. a) Explain the insertion operation linked list. How nodes are inserted after a specified node ?

(OR)

- b) What are the applications of linked list in dynamic storage management ?



12. a) Write an algorithm for Push and Pop operations on Stack using Linked list.
(OR)

b) What is a DeQueue ? Explain its operation with example.

13. a) Explain the tree traversal techniques with an example.
(OR)

b) How to insert and delete an element into a binary search tree and write down the code for the insertion routine with an example.

14. a) Explain depth first and breadth first traversal.
(OR)

b) Explain the various applications of Graphs.

15. a) Write an algorithm to implement selection sort with suitable example.
(OR)

b) Write an algorithm for binary search with suitable example.

PART - C

(1×15=15 Marks)

16. a) There are 'N' numbers of balls in the box. The colours of the balls are red and blue. You are requested to stack the balls in the bottom sealed basket one by one. The order of placing the balls is two consecutive red balls followed by the two consecutive blue balls. Later, Create two empty queues Q1 and Q2. Remove the last inserted ball from the basket and place it in Q1. Similarly remove the next ball from the basket and insert in Q2. Develop a program to repeat this process until the basket is empty and also print the colour of the balls in both queues.

(OR)

b) Implement a priority queue using linked list.

REVIEW SHEET

After Completion of syllabus
Faculty experience in handling / covering syllabus
Unit I : Students to be given more practice to write algorithms - DS Implementation to carry various operations to be given more practice. Position coverage can be made as planned.
Unit II : Stack, Queue implementation can be covered & complexity level is less. practice on Queue varieties & applications to be given more practice. Few additional hours to practice exercise is required.
Unit III : Covering this unit requires few additional hours. More focus on BST, AVL, B Trees, B+ trees is required.
Unit IV : Coverage can be made as planned. Comparatively simple Unit.
Unit V : Coverage can be made as planned. Sorting techniques should be focused (ie practice to write algorithms),
Difficulties (if any) In general more practice to perform application based operations on various data structures to be given practice. Book exercise based problem solving practice to be given.
Feedback on University Question Paper Standard questions were asked. From internal assessment question paper, part B questions were asked. virtual lab session based 1 part C question was asked.
 SIGNATURE OF STAFF  HOD/CSE