DATA STRUCTURES [CS302PC]

COURSE PLANNER

I. CourseOverview:

This course introduces the core principles and techniques for Data structures. Students will gain experience in how to keep a data in an ordered fashion in the computer. Students can improve their programming skills using Data Structures Concepts through C.

II. Prerequisite:

A course on "Programming for Problem Solving".

III. CourseObjective:

S. No	Objective
1	Exploring basic data structures such as stacks and queues.
2	Introduces a variety of data structures such as hash tables, search trees, tries, heaps, graphs
3	Introduces sorting and pattern matching algorithms

IV. CourseOutcome:

Course	CO. No.	Course Outcomes (CO)	Knowledge Level (Blooms Level)
Data Structures	CO1	***Ability*** to select the data structures that efficiently model the information in a problem.	**L4:Analysis**
	CO2	***Ability*** to assess efficiency trade-offs among different data structure implementations or combinations.	**L4:Analysis**
	CO3	***Implement*** and know the application of algorithms for sorting and pattern matching.	**L5: Synthesis**
	CO4	***Design*** programs using a variety of data structures, including hash tables, binary and general tree structures, search trees, tries, heaps, graphs, and AVL-trees.	**L6:Create**

V. How program outcomes areassessed:

Program Outcomes (PO)		Level	Proficiency assessed by
PO1	**Engineeering knowledge:** Apply the knowledge of Mathematics, science, engineering fundamentals and an engineering specialization to the solution of	2.5	Assignments, Tutorials, Mock

	complex engineering problems.		
PO2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	2.5	Assignments, Tutorials, Mock Tests
PO3	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.	2.75	Assignments, Tutorials, Mock Tests
PO4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.	2	Assignments, Tutorials, Mock Tests
PO5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.	-	--
PO6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.	-	--
PO7	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.	-	--
PO8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice	-	--
PO9	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	1.75	Assignments, Tutorials, Mock Tests
PO10	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and	-	--

	design documentation, make effective presentations, and give and receive clear instructions.		
PO11	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.	-	--
PO12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change	-	--

1:Slight(Low) **2:Moderate(Medium)** **3:Substantial (High)** **- :Non e**

VI. How program specific outcomes areassessed:

Program Specific Outcomes (PSO)		**Level**	**Proficiency assessed by**
PSO1	**Software Development and Research Ability:** Ability to understand the structure and development methodologies of software systems. Possess professional skills and knowledge of software design process. Familiarity and practical competence with a broad range of programming language and open source platforms. Use knowledge in various domains to identify research gaps and hence to provide solution to new ideas and innovations.	1.5	Lectures, Assignme nts, Tutorials, Mock Tests
PSO2	**Foundation of mathematical concepts:** Ability to apply the acquired knowledge of basic skills, principles of computing, mathematical foundations, algorithmic principles, modeling and design of computer- based systems in solving real world engineering Problems.	2.5	Lectures, Assignments, Tutorials, Mock Tests
PSO3	**Successful Career:** Ability to update knowledge continuously in the tools like Rational Rose, MATLAB, Argo UML, R Language and technologies like Storage, Computing, Communication to meet the industry requirements in creating innovative career paths for immediate employment and for higher studies.	2.5	Lectures, Assignments
1: Slight (Low)	**2: Moderate (Medium)**	**3: Substantial (High)**	**- : None**

VII. Syllabus:

UNIT – I

Introduction to Data Structures, abstract data types, Linear list – singly linked list implementation, insertion, deletion and searching operations on linear list, Stacks-Operations, array and linked representations of stacks, stack applications, Queues-operations, array and linked representations.

UNIT - II

Dictionaries: linear list representation, skip list representation, operations - insertion, deletion and searching.

Hash Table Representation: hash functions, collision resolution-separate chaining, open addressing-linear probing, quadratic probing, double hashing, rehashing, extendible hashing.

UNIT - III

Search Trees: Binary Search Trees, Definition, Implementation, Operations- Searching, Insertion and Deletion, AVL Trees, Definition, Height of an AVL Tree, Operations – Insertion, Deletion and Searching, Red –Black, Splay Trees.

UNIT - IV

Graphs: Graph Implementation Methods. Graph Traversal Methods.

Sorting: Heap Sort, External Sorting- Model for external sorting, Merge Sort.

UNIT - V

Pattern Matching and Tries: Pattern matching algorithms-Brute force, the Boyer –Moore algorithm, the Knuth-Morris-Pratt algorithm, Standard Tries, Compressed Tries, Suffix tries.

TEXT BOOKS:

1. Fundamentals of Data Structures in C, 2nd Edition, E. Horowitz, S. Sahni andSusan Anderson Freed, UniversitiesPress.
2. Data Structures using C – A. S. Tanenbaum, Y. Langsam, and M.J.Augenstein, PHI/PearsonEducation.

REFERENCE BOOKS:

1. Data Structures: A Pseudocode Approach with C, 2nd Edition, R. F. Gilberg and B.A. Forouzan, Cengage Learning.

NPTEL web course:

https://onlinecourses.nptel.ac.in

RELEVANT SYLLABUS FORGATE:

Structured programming with Pascal/C including recursion; arrays, stacks, strings, queues, lists, trees, sets and graphs; algorithm for tree and graphs traversals, connected component, spanning trees, shortest paths; hashing, sorting and searching algorithm design and analysistechniques.

RELEVANT SYLLABUS FOR IES: -NA-

VIII. CoursePlan:

Lecture No.	Week	Unit No.	Topics to be covered	Link for PPT	Link for PDF	Course Learning Outcome	Teaching Methodology	Reference
1	1		Object Based Education(OBE) Orientation	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	**Understand** OBE	PPT	Fundamentals of Data structures in c, 2nd Edition, E.Horowitz, S.Sahani and Susan Anderson - Freed, Universities Press
2		1	**UNIT-I:** Introduction to Data Structures	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	**Define** DS	Talk & Chalk, Discussion	
3			Abstract Data Type	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	**Know** about ADT	Talk & Chalk, Discussion	
4			Introduction to Linear DS	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	**Understand** linear DS	Talk & Chalk, Discussion	
5	2		Singly Linked Lists-Operations-Insertion, Deletion	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	**Analyze** operations of linear DS	Talk & Chalk, Discussion	
6			Singly Linked Lists-Operations-Implementation, Searching	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	**Analyze** operations of linear DS	Talk & Chalk, Discussion	
7			**Double Linked List **Circular Linked List	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	**Know** about DLL & CLL	PPT	

8			Stack-definition, operations	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	**Understand** Stack	Talk & Chalk, Discussion	
9			Stack-Array & Linked Representations	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	**Analyze** Stack representations	Talk & Chalk, Discussion	
10	3		Stack Applications	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	**Define** applications of stack	Talk & Chalk, Discussion	
11			Queue-definition, operations	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	**Understand** Queue	Talk & Chalk, Discussion	
12			Queue-Array & Linked Representations	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	https://drive.google.com/drive/folders/1Y0LJX0IONQorhc9HGsCiAx1Aoi6rl9cV	**Analyze** Queue representations	Talk & Chalk, Discussion	
13			***Mock Test #1***					
14	4	2	**UNIT II:**Dictionaries-Introduction	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXpU7lQRaU-_7z77tF0wd	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXpU7lQRaU-_7z77tF0wd	**Understand** Dictionaries	Talk & Chalk, Discussion	
15			Dictionaries-Linear List Representation	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXpU7lQRaU-_7z77tF0wd	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXpU7lQRaU-_7z77tF0wd	**Analyze** representation of dictionaries	Talk & Chalk, Discussion	
16			***Bridge Class #1***	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXp	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXpU7lQ		Discussion	Fundamentals of Data structures in c, 2nd

				U7lQRaU-_7z77tF0wd	RaU-_7z77tF0wd			Edition, E.Horowitz, S.Sahani and Susan Anderson - Freed, Universities Press
17	5		Dictionaries-Skip List Representation	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXpU7lQRaU-_7z77tF0wd	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXpU7lQRaU-_7z77tF0wd	**Analyze** representation of dictionaries	Talk & Chalk, Discussion	
18			Dictionaries-Operations:Insertion, Deletion & Searching	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXpU7lQRaU-_7z77tF0wd	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXpU7lQRaU-_7z77tF0wd	**Define** operations of dictionaries	Talk & Chalk, Discussion	
19			Hashing-Introduction, Hash table representation	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXpU7lQRaU-_7z77tF0wd	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXpU7lQRaU-_7z77tF0wd	**Understand** Hashing	Talk & Chalk, Discussion	
20	6		***Bridge Class #2***	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXpU7lQRaU-_7z77tF0wd	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXpU7lQRaU-_7z77tF0wd		Discussion	
21			Hash functions	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXpU7lQRaU-_7z77tF0wd	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXpU7lQRaU-_7z77tF0wd	**Understand** Hashing	Talk & Chalk, Discussion	
22			Collision resolutions-Separate chaining	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXpU7lQRaU-_7z77tF0wd	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXpU7lQRaU-_7z77tF0wd	**Understand** Hashing	Talk & Chalk, Discussion	
23	7		Open addressing-linear & quadratic probing	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXpU7lQRaU-_7z77tF0wd	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXpU7lQRaU-_7z77tF0wd	**Understand** Hashing	Talk & Chalk, Discussion	
24			double hashing and rehashing,extendible hashing	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXp	https://drive.google.com/drive/folders/1j8mYwBfxGHTDXpU7lQ	**Understand** Hashing	Talk & Chalk, Discussion	

				U7lQRaU-_7z77tF0wd	RaU-_7z77tF0wd			
25		3	**UNIT III:**Introduction about Trees	https://drive.google.com/drive/folders/1fPHDNxf-y6sjUD1WY4nfB2JoFaMYkWEr	https://drive.google.com/drive/folders/1fPHDNxf-y6sjUD1WY4nfB2JoFaMYkWEr	**Know** Trees	Talk & Chalk, Discussion	
26			Binary search tree definition,implementation	https://drive.google.com/drive/folders/1fPHDNxf-y6sjUD1WY4nfB2JoFaMYkWEr	https://drive.google.com/drive/folders/1fPHDNxf-y6sjUD1WY4nfB2JoFaMYkWEr	**Understand** BST	Talk & Chalk, Discussion	
27	8		Binary search tree-operations:Searching,Insertion & Deletion	https://drive.google.com/drive/folders/1fPHDNxf-y6sjUD1WY4nfB2JoFaMYkWEr	https://drive.google.com/drive/folders/1fPHDNxf-y6sjUD1WY4nfB2JoFaMYkWEr	**Analyze** BST operations	Talk & Chalk, Discussion	
28			Binary search tree-operations:Searching,Insertion & Deletion	https://drive.google.com/drive/folders/1fPHDNxf-y6sjUD1WY4nfB2JoFaMYkWEr	https://drive.google.com/drive/folders/1fPHDNxf-y6sjUD1WY4nfB2JoFaMYkWEr	**Analyze** BST operations	Talk & Chalk, Discussion	
29			***Bridge Class #3***	https://drive.google.com/drive/folders/1fPHDNxf-y6sjUD1WY4nfB2JoFaMYkWEr	https://drive.google.com/drive/folders/1fPHDNxf-y6sjUD1WY4nfB2JoFaMYkWEr		Discussion	
30			AVL trees-Definition and height of an AVL tree	https://drive.google.com/drive/folders/1fPHDNxf-y6sjUD1WY4nfB2JoFaMYkWEr	https://drive.google.com/drive/folders/1fPHDNxf-y6sjUD1WY4nfB2JoFaMYkWEr	**Understand** AVL	Talk & Chalk, Discussion	
31			AVL tree-operations:Searching,Insertion & Deletion	https://drive.google.com/drive/folders/1fPHDNxf-y6sjUD1WY4nfB2JoFaMYkWEr	https://drive.google.com/drive/folders/1fPHDNxf-y6sjUD1WY4nfB2JoFaMYkWEr	**Analyze** AVL operations	Talk & Chalk, Discussion	
32	9	3	Red-Black trees	https://drive.google.com/drive/folders/1fPHDNxf-	https://drive.google.com/drive/folders/1fPHDNxf-	**Understand** Red-Black tree	Talk & Chalk, Discussion	Fundamentals of Data structures in c, 2nd

				y6sjUD1WY4nfB2JoFaMYkWEr	y6sjUD1WY4nfB2JoFaMYkWEr			Edition, E.Horowitz, S.Sahani and Susan Anderson - Freed, Universities Press
33			Splay trees	https://drive.google.com/drive/folders/1fPHDNxf-y6sjUD1WY4nfB2JoFaMYkWEr	https://drive.google.com/drive/folders/1fPHDNxf-y6sjUD1WY4nfB2JoFaMYkWEr	**Understand** Splay tree	Talk & Chalk, Discussion	
34			** B-trees	https://drive.google.com/drive/folders/1fPHDNxf-y6sjUD1WY4nfB2JoFaMYkWEr	https://drive.google.com/drive/folders/1fPHDNxf-y6sjUD1WY4nfB2JoFaMYkWEr	**Understand** B-tree	Talk & Chalk	
35	10	4	**UNIT IV:** Graphs-Introduction,Definition, Terminology	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW8	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW8	**Define** Graphs	Talk & Chalk, Discussion	
36			Graph implementation methods	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW9	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW9	**Analyze** Graph implementation	Talk & Chalk, Discussion	
37			Graph traversals-DFS	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW10	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW10	**Implement** DFS	Talk & Chalk, Discussion	
38	11		Graph traversals-BFS	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW11	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW11	**Implement** BFS	Talk & Chalk, Discussion	
39			***Seminars by students***	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW12	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW12			
40			Sorting-introduction	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHS	**Define** Sorting	Talk & Chalk, Discussion	

				6gHSMef9_oMVfkW13	Mef9_oMVfkW13			
41	12		Heap sort	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW14	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW14	**Understand** Heap Sort	Talk & Chalk, Discussion	
42			External sorting-models of external sorting	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW15	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW15	**Understand** External Sort	Talk & Chalk, Discussion	
43			External sorting-models of external sorting	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW16	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW16	**Understand** External Sort	Talk & Chalk, Discussion	
44			Merge sort	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW17	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW17	**Understand** Merge Sort	Talk & Chalk, Discussion	Fundamentals of Data structures in c, 2nd Edition, E.Horowitz, S.Sahani and Susan Anderson - Freed, Universities Press
45	13		**Insertion sort **Selection sort	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW18	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW18	**Understand** basic sort	PPT	
46			***Bridge Class #4***	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW19	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW19		Discussion	
47			***Mock Test #2***	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW20	https://drive.google.com/drive/folders/1bCpr8evIwSHclA6x6gHSMef9_oMVfkW20			
48		5	**UNIT V:** Pattern matching algorithm-introduction	https://drive.google.com/drive/folders/1lhY	https://drive.google.com/drive/folders/1lhYmV	**Define** Pattern Matching	Talk & Chalk, Discussion	

				mV2njVh4pMdP-PvBZW_0zMZ8dfEZP	2njVh4pMdP-PvBZW_0zMZ8dfEZP		
49	14		Brute force	https://drive.google.com/drive/folders/1lhYmV2njVh4pMdP-PvBZW_0zMZ8dfEZP	https://drive.google.com/drive/folders/1lhYmV2njVh4pMdP-PvBZW_0zMZ8dfEZP	**Implement** Brute force	Talk & Chalk, Discussion
50			Boyer –Moore algorithm	https://drive.google.com/drive/folders/1lhYmV2njVh4pMdP-PvBZW_0zMZ8dfEZP	https://drive.google.com/drive/folders/1lhYmV2njVh4pMdP-PvBZW_0zMZ8dfEZP	**Implement** Boyer-Moore	Talk & Chalk, Discussion
51			Knuth-Morris-Pratt algorithm	https://drive.google.com/drive/folders/1lhYmV2njVh4pMdP-PvBZW_0zMZ8dfEZP	https://drive.google.com/drive/folders/1lhYmV2njVh4pMdP-PvBZW_0zMZ8dfEZP	**Implement** Knuth-Morris-Pratt	Talk & Chalk, Discussion
52			Standard Tries	https://drive.google.com/drive/folders/1lhYmV2njVh4pMdP-PvBZW_0zMZ8dfEZP	https://drive.google.com/drive/folders/1lhYmV2njVh4pMdP-PvBZW_0zMZ8dfEZP	**Analyze** Tries	Talk & Chalk, Discussion
53	15		Compressed Tries	https://drive.google.com/drive/folders/1lhYmV2njVh4pMdP-PvBZW_0zMZ8dfEZP	https://drive.google.com/drive/folders/1lhYmV2njVh4pMdP-PvBZW_0zMZ8dfEZP	**Analyze** Tries	Talk & Chalk, Discussion
54			Suffix tries	https://drive.google.com/drive/folders/1lhYmV2njVh4pMdP-PvBZW_0zMZ8dfEZP	https://drive.google.com/drive/folders/1lhYmV2njVh4pMdP-PvBZW_0zMZ8dfEZP	**Analyze** Tries	Talk & Chalk, Discussion
55			Tries	https://drive.google.com/driv	https://drive.google.com/drive/	**Analyze** Tries	Talk & Chalk, Discussion

				e/folders/1lhYmV2njVh4pMdP-PvBZW_0zMZ8dfEZP	folders/1lhYmV2njVh4pMdP-PvBZW_0zMZ8dfEZP			
56	16		***Bridge Class #5***	https://drive.google.com/drive/folders/1lhYmV2njVh4pMdP-PvBZW_0zMZ8dfEZP	https://drive.google.com/drive/folders/1lhYmV2njVh4pMdP-PvBZW_0zMZ8dfEZP		Discussion	

IX. Mapping course outcomes leading to the achievement of program outcomesand program specificoutcomes:

Course Outcomes	Program Outcomes (PO)												Program Specific Outcomes (PSO)		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	3	2	2	-	-	-	-	3	-	-	-	2	1	3
CO2	3	3	3	2	-	-	-	-	2	-	-	-	1	3	2
CO3	2	2	3	2	-	-	-	-	1	-	-	-	2	3	2
CO4	3	2	3	2	-	-	-	-	1	-	-	-	1	3	3
AVG	**2.5**	**2.5**	**2.75**	**2**	-	-	-	-	**1.75**	-	-	-	**1.5**	**2.5**	**2.5**

1: Slight (Low) **2: Moderate (Medium)** **3: Substantial (High)** **- : None**

X. QUESTION BANK (JNTUH)

UNIT I

Long Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome
1	Explain the various operations that can be performed on different Data Structures?	Understanding	2
2	Define Queue, how it is different from stack and how is it implemented?	Remembering	1
3	State the difference between queues and linked list?	Understanding	2
4	Define Stack? Explain about application of stack?	Remembering	1
5	Explain about operations in single linked list?	Understanding	2
6	Define the implementation of Queue with array and linked list?	Remembering	1

Short Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome

1	Explain how an array different from linked list?	Understanding	2
2	Define Stack and where it can be used?	Remembering	1
3	Explain about queue operations in brief?	Understanding	2
4	Define LIFO?	Remembering	1
5	Which data structure is used for dictionary and spell checker?	Remembering	1

UNIT II

Long Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome
1	Distinguish between double hashing and rehashing?	Analyzing	4
2	Explain about different representations of Dictionaries?	Understanding	2
3	Explain the various operations of Dictionaries?	Understanding	2
4	Distinguish between double hashing, rehashing, and extendible hashing?	Analyzing	4
5	Explain about linear probing and quadratic probing?	Understanding	2

Short Answer Questions-

S.No	Question	Blooms Taxonomy Leve	Course Outcom
1	Define hash functions?	Remembering	1
2	Define Dictionaries?	Remembering	1
3	List the applications of Dictionaries?	Analyzing	4
4	List the applications of hashing?	Analyzing	4

UNIT III

Long Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome
1	What is Binary search Tree? Explain about operations on Binary search tree?	Remembering	1
2	Describe in brief about array and linked representations of binary search tree?	Understanding	2
3	Describe a procedure to insert and delete an element into a AVL Tree?	Understanding	2
4	Describe a procedure to search an element in a AVL Tree?	Understanding	2
5	Explain about Red-Black tree and Splay tree with example?	Understanding	2

Short Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome
1	Define searching in binary search tree with an example?	Remembering	1
2	Explain the height of an AVL Tree?	Understanding	2
3	Describe Splay tree with example?	Understanding	2
4	Define AVL tree. Explain the acceptable balancing factor of	Remembering	1

	AVL tree?		
5	Distinguish between BST and AVL tree?	Analyzing	4

UNIT IV

Long Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome
1	Explain the graph traversal methods?	Understanding	2
2	Explain the time complexity of merge sort in best, worst and average case?	Understanding	2
3	Illustrate the concept of Merge sort with example?	Understanding	2
4	Implement Merge sort using C?	Applying	3
5	Illustrate the concept of Heap sort with example?	Understanding	2

Short Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome
1	Write a short note on representation of Graphs?	Understanding	2
2	Distinguish between graph and tree?	Analyzing	4
3	Define sorting? Explain about external Sorting?	Remembering	1
4	Describe the concept of graph traversals with an example?	Understanding	2
5	Explain the algorithm of Merge sort?	Understanding	2

UNIT V

Long Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome
1	Explain about Knuth-Morris-Pratt algorithm with example?	Understanding	2
2	Explain about Pattern matching algorithms and its applications?	Understanding	2
3	Explain about Compressed Tries and Suffix tries?	Understanding	2
4	Distinguish between Standard Tries and Compressed Tries?	Analyzing	4
5	Distinguish between Suffix tries and Compressed Tries?	Analyzing	4

Short Answer Questions-

S.No	Question	Blooms Taxonomy Level	Course Outcome
1	Define pattern matching?	Remembering	1
2	Explain short note on Pattern matching algorithms?	Understanding	2
3	Define Compressed Tries?	Remembering	1
4	Define Standard Tries?	Remembering	1
5	Explain about Knuth-Morris-Pratt algorithm?	Understanding	2

OBJECTIVE QUESTIONS: CHOOSE CORRECT OPTION

UNIT I:

1. Which of the following points is/are true about Linked List data structure when it is compared with array

a. Arrays have better cache locality that can make them better in terms of performance.

b. It is easy to insert and delete elements in Linked List
c. Random access is not allowed in a typical implementation of Linked Lists
d. All of the above

2. What is the output of following function for start pointing to first node of following linked list? 1->2->3->4->5->6

```
void fun(struct node* start)
{
  if(start == NULL)
    return;
  printf("%d  ", start->data);

  if(start->next != NULL )
    fun(start->next->next);
  printf("%d  ", start->data);
}
```

a. 1 4 6 6 4 1
b. 1 3 5 1 3 5
c. 1 2 3 5
d. 1 3 5 5 3 1

3. In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is
a. log 2 n
b. n/2
c. log 2 n – 1
d. n

4. Suppose each set is represented as a linked list with elements in arbitrary order. Which of the operations among union, intersection, membership, cardinality will be the slowest?
a. union only
b. intersection, membership
c. membership, cardinality
d. union, intersection

5. Which one of the following is an application of Stack Data Structure?
a. Managing function calls
b. The stock span problem
c. Arithmetic expression evaluation
d. All of the above

6. Which of the following is true about linked list implementation of stack?
a. In push operation, if new nodes are inserted at the beginning of linked list, then in pop operation, nodes must be removed from end.
b. In push operation, if new nodes are inserted at the end, then in pop operation, nodes must be removed from the beginning.
c. Both of the above
d. None of the above

7. The following postfix expression with single digit operands is evaluated using a stack:
8 2 3 ^ / 2 3 * + 5 1 * -

Note that ^ is the exponentiation operator. The top two elements of the stack after the first * is evaluated are:

a. 6,1
b. 5,7
c. 3,2
d. 1,5

8. Let S be a stack of size n >= 1. Starting with the empty stack, suppose we push the first n natural numbers in sequence, and then perform n pop operations. Assume that Push and Pop operation take X seconds each, and Y seconds elapse between the end of one such stack operation and the start of the next operation. For m >= 1, define the stack-life of m as the time elapsed from the end of Push(m) to the start of the pop operation that removes m from S. The average stack-life of an element of this stack is

a. n(X+ Y)
b. 3Y + 2X
c. n(X + Y)-X
d. Y + 2X

9. Which one of the following is an application of Queue Data Structure?

a. When a resource is shared among multiple consumers.
b. When data is transferred asynchronously (data not necessarily received at same rate as sent) between two processes
c. Load Balancing
d. All of the above

10. How many stacks are needed to implement a queue. Consider the situation where no other data structure like arrays, linked list is available to you.

a. 1
b. 2
c. 3
d. 4

11. How many queues are needed to implement a stack. Consider the situation where no other data structure like arrays, linked list is available to you.

a. 1
b. 2
c. 3
d. 4

12. A priority queue can efficiently implemented using which of the following data structures? Assume that the number of insert and peek (operation to see the current highest priority item) and extraction (remove the highest priority item) operations are almost same.

a. Array
b. Linked List
c. Heap Data Structures like Binary Heap, Fibonacci Heap
d. None of the above

13. Which of the following is true about linked list implementation of queue?

a. In push operation, if new nodes are inserted at the beginning of linked list, then in pop operation, nodes must be removed from end.
b. In push operation, if new nodes are inserted at the end, then in pop operation, nodes must be removed from the beginning.
c. Both of the above

d. None of the above

UNIT II:

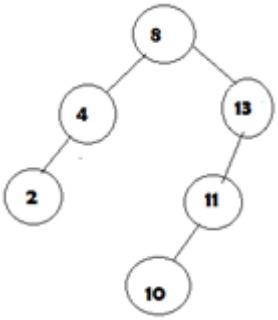
1. Which of the following is the efficient data structure for searching words in dictionaries?
   a) BST
   b) Linked List
   c) Balancded BST
   d) Trie
2. What is a hash table?
   a) A structure that maps values to keys
   b) A structure that maps keys to values
   c) A structure used for storage
   d) A structure used to implement stack and queue
3. If several elements are competing for the same bucket in the hash table, what is it called?
   a) Diffusion
   b) Replication
   c) Collision
   d) None of the mentioned
4. What is a hash function?
   a) A function has allocated memory to keys
   b) A function that computes the location of the key in the array
   c) A function that creates an array
   d) None of the mentioned
5. What can be the techniques to avoid collision?
   a) Make the hash function appear random
   b) Use the chaining method
   c) Use uniform hashing
   d) All of the mentioned
6. What is simple uniform hashing?
   a) Every element has equal probability of hashing into any of the slots
   b) A weighted probabilistic method is used to hash elements into the slots
   c) All of the mentioned
   d) None of the mentioned
7. Double hashing is one of the best methods available for open addressing.
   a) True
   b) False
8. What is the hash function used in Double Hashing?
   a) (h1(k) – i*h2(k))mod m
   b) h1(k) + h2(k)
   c) (h1(k) + i*h2(k))mod m
   d) (h1(k) + h2(k))mod m
9. What are the values of h1(k) and h2(k) in the hash function?
   a) h1(k) = m mod k
      h2(k) = 1+ (m' mod k)
   b) h1(k) = 1 + (m mod k)
      h2(k) = m' mod k
   c) h1(k) = 1+ (k mod m)
      h2(k) = k mod m

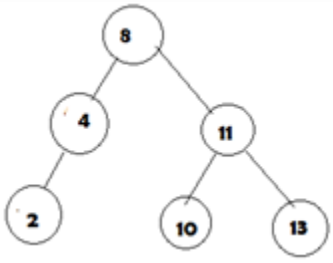
d) h1(k) = k mod m
h2(k) = 1+ (k mod m')

10. Which of the following schemes does quadratic probing come under?
    a) rehashing
    b) extended hashing
    c) separate chaining
    d) open addressing
11. What kind of deletion is implemented by hashing using open addressing?
    a) active deletion
    b) standard deletion
    c) lazy deletion
    d) no deletion
12. Which of the following problems occur due to linear probing?
    a) Primary collision
    b) Secondary collision
    c) Separate chaining
    d) Extendible hashing
13. How many probes are required on average for insertion and successful search?
    a) 4 and 10
    b) 2 and 6
    c) 2.5 and 1.5
    d) 3.5 and 1.5

UNIT III:

1. What is an AVL tree?
    a) a tree which is balanced and is a height balanced tree
    b) a tree which is unbalanced and is a height balanced tree
    c) a tree with three children
    d) a tree with atmost 3 children
2. Which of the below diagram is following AVL tree property?
    i.





a) only i
b) only i and ii
c) only ii
d) none of the mentioned

3. What is the maximum height of an AVL tree with p nodes?
a) p
b) log(p)
c) log(p)/2
d) p/2

4. Given an empty AVL tree, how would you construct AVL tree when a set of numbers are given without performing any rotations?
a) just build the tree with the given input
b) find the median of the set of elements given, make it as root and construct the tree
c) use trial and error
d) use dynamic programming to build the tree

5. What maximum difference in heights between the leafs of a AVL tree is possible?
a) log(n) where n is the number of nodes
b) n where n is the number of nodes
c) 0 or 1
d) atmost 1

6. Why to prefer red-black trees over AVL trees?
a) Because red-black is more rigidly balanced
b) AVL tree store balance factor in every node which costs space
c) AVL tree fails at scale
d) Red black is more efficient

7. Which of the following is false about a binary search tree?
a) The left child is always lesser than its parent
b) The right child is always greater than its parent
c) The left and right sub-trees should also be binary search trees
d) None of the mentioned

8. What is the speciality about the inorder traversal of a binary search tree?
a) It traverses in a non increasing order
b) It traverses in an increasing order
c) It traverses in a random fashion
d) None of the mentioned

9. What are the worst case and average case complexities of a binary search tree?
a) O(n), O(n)
b) O(logn), O(logn)

c) O(logn), O(n)
d) O(n), O(logn)
10. What are the conditions for an optimal binary search tree and what is its advantage?
a) The tree should not be modified and you should know how often the keys are accessed, it improves the lookup cost
b) You should know the frequency of access of the keys, improves the lookup time
c) The tree can be modified and you should know the number of elements in the tree before hand, it improves the deletion time
d) None of the mentioned

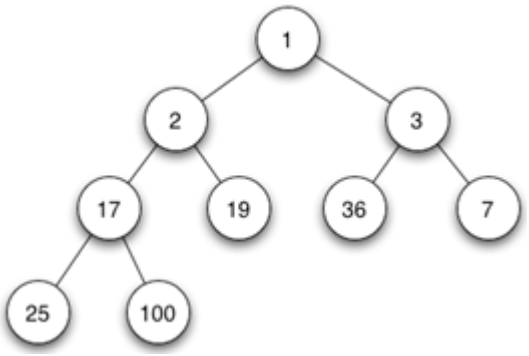
UNIT IV:

1. Which of the following sorting algorithms can be used to sort a random linked list with minimum time complexity?
a. Insertion Sort
b. Quick Sort
c. Heap Sort
d. Merge Sort

2. In a max-heap, element with the greatest key is always in the which node?
a) Leaf node
b) First node of left sub tree
c) root node
d) First node of right sub tree

3. Heap can be used as _____
a) Priority queue
b) Stack
c) A decreasing order array
d) None of the mentioned

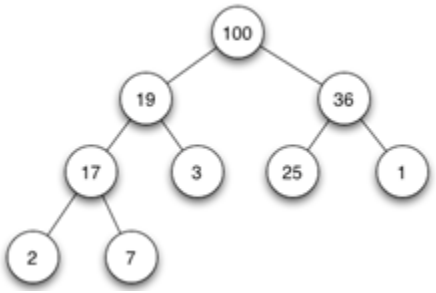
4. If we implement heap as min-heap , deleting root node (value 1)from the heap. What would be the value of root node after second iteration if leaf node (value 100) is chosen to replace the root at start.



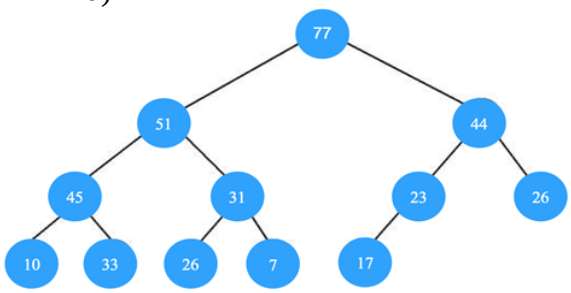
a) 2
b) 100
c) 17
d) 3

5. If we implement heap as maximum heap , adding a new node of value 15 to the left most node of right subtree . What value will be at leaf nodes of the right subtree of the heap.
a) 15 and 1

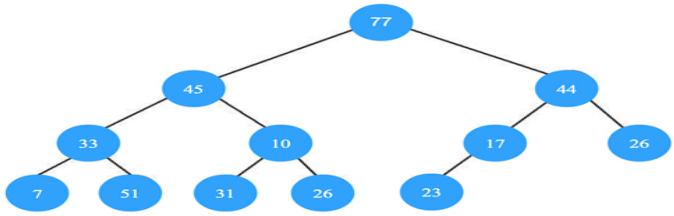
b) 25 and 1
c) 3 and 1
d) 2 and 3



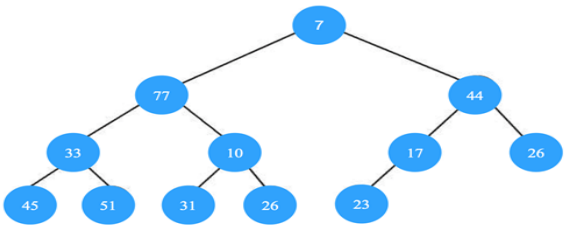
6. Descending priority queue can be implemented using _____
   a) max heap
   b) min heap
   c) min-max heap
   d) trie
7. Min heap can be used to implement selection sort.
   a) True
   b) False
8. Which of the following is the valid min heap?
   a)



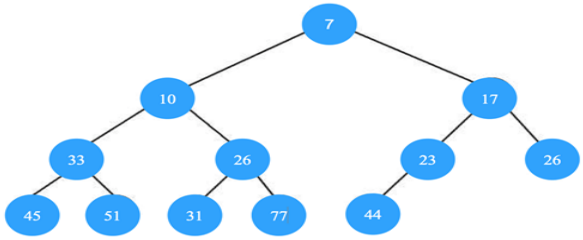
b)



c)



d)



9. Which one of the following array elements represents a binary min heap?
   a) 12 10 8 25 14 17
   b) 8 10 12 25 14 17
   c) 25 17 14 12 10 8
   d) 14 17 25 10 12 8
10. In a binary min heap containing n elements, the largest element can be found in _____ time.
   a) O(n)
   b) O(nlogn)
   c) O(logn)
   d) O(1)

UNIT V:

1. Trie is also known as _____
   a) Digital Tree
   b) Treap
   c) Binomial Tree
   d) 2-3 Tree
2. Which of the following special type of trie is used for fast searching of the full texts?
   a) Ctrie
   b) Hash tree
   c) Suffix tree
   d) T tree
3. Which of the following is true about the trie?
   a) root is letter a
   b) path from root to the leat yields the string
   c) children of nodes are randomly ordered
   d) each node stores the associated keys
4. Auto complete and spell checkers can be implemented efficiently using the trie.
   a) True
   b) False

GATE RELATED QUESTIONS

1. A single array A[1..MAXSIZE] is used to implement two stacks. The two stacks grow from opposite ends of the array. Variables top1 and top2 (topl< top 2) point to the location of the topmost element in each of the stacks. If the space is to be used efficiently, the condition for "stack full" is

a. (top1 = MAXSIZE/2) and (top2 = MAXSIZE/2+1)
b. top1 + top2 = MAXSIZE
c. (top1= MAXSIZE/2) or (top2 = MAXSIZE)
d. top1= top2 -1

2. Assume that the operators +, -, × are left associative and ^ is right associative. The order of precedence (from highest to lowest) is ^, x , +, -. The postfix expression corresponding to the infix expression a + b × c - d ^ e ^ f is

a. abc × + def ^ ^ -
b. abc × + de ^ f ^ -
c. ab + c × d - e ^ f ^
d. - + a × bc ^ ^ def

3. Suppose a circular queue of capacity (n – 1) elements is implemented with an array of n elements. Assume that the insertion and deletion operation are carried out using REAR and FRONT as array index variables, respectively. Initially, REAR = FRONT = 0. The conditions to detect queue full and queue empty are

a. Full: (REAR+1) mod n == FRONT, empty: REAR == FRONT
b. Full: (REAR+1) mod n == FRONT, empty: (FRONT+1) mod n == REAR
c. Full: REAR == FRONT, empty: (REAR+1) mod n == FRONT
d. Full: (FRONT+1) mod n == REAR, empty: REAR == FRONT

4. A Priority-Queue is implemented as a Max-Heap. Initially, it has 5 elements. The level-order traversal of the heap is given below: 10, 8, 5, 3, 2 Two new elements "1' and "7' are inserted in the heap in that order. The level-order traversal of the heap after the insertion of the elements is:

a. 10, 8, 7, 5, 3, 2, 1
b. 10, 8, 7, 2, 3, 1, 5
c. 10, 8, 7, 1, 2, 3, 5
d. 10, 8, 7, 3, 2, 1, 5

5. If arity of operators is fixed, then which of the following notations can be used to parse expressions without parentheses? a) Infix Notation (Inorder traversal of a expression tree) b) Postfix Notation (Postorder traversal of a expression tree) c) Prefix Notation (Preorder traversal of a expression tree)

a. b and c
b. Only b
c. a, b and c
d. None of them

6. Level of a node is distance from root to that node. For example, level of root is 1 and levels of left and right children of root is 2. The maximum number of nodes on level i of a binary tree is

In the following answers, the operator '^' indicates power.

a. 2^(i)-1
b. 2^i
c. 2^(i+1)

d. 2^[(i+1)/2]

7. The height of a binary tree is the maximum number of edges in any root to leaf path. The maximum number of nodes in a binary tree of height h is:

a. 2^h -1
b. 2^(h-1) – 1
c. 2^(h+1) -1
d. 2*(h+1)

8. A complete n-ary tree is a tree in which each node has n children or no children. Let I be the number of internal nodes and L be the number of leaves in a complete n-ary tree. If L = 41, and I = 10, what is the value of n?

a. 6
b. 3
c. 4
d. 5

9. The number of leaf nodes in a rooted tree of n nodes, with each node having 0 or 3 children is:

a. n/2
b. (n-1)/3
c. (n-1)/2
d. (2n+1)/3

10. The maximum number of binary trees that can be formed with three unlabeled nodes is

a. 1
b. 5
c. 4
d. 3

WEBSITES' ADDRESSES:

1. http://www.dreamincode.net/forums/forum/48-c-tutorials/
2. http://nptel.iitm.ac.in/video.php?subjectId=106102064
3. http://www.tutorialspoint.com/cplusplus/cpp_data_structures.htm
4. http://www.sourcecodesworld.com/source/BrowseCategory.asp?CatId=33

EXPERT DETAILS:

1. Dr. Naveen Garg from IIT DELHI.
2. Dr.Pradip Das from IIT Guwahati
3. Dr.Padmanabam from JNTUH.

LIST OF TOPICS FOR STUDENTS' SEMINARS:

1. Applications of Trees
2. Comparative study of all the data structures.
3. Applications of Graphs.
4. Comparative study of all the types of trees.

CASE STUDIES / SMALL PROJECTS:

Implement the following programs using C

1. Concatenation of two Single Linked List
2. Removing duplicate element of linked list
3. Queue using two stacks
4. Splay trees