MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

MUST KNOW CONCEPTS

MKC

MCA

2021-2022

Subject Code/Name : **21CAB02 - Data Structures And Algorithms**

Year/Sem/Sec : **I / I / -**

S.No	Term	Notation (Symbol)	Concept/Definition/Meaning/Units/Equation /Expression	Units
			Unit I: Linear Data Structures	
1	Data Structure	--	Data Structures is defined as the way of organizing all data items that consider not only the elements stored but also stores the relationship between the elements.	I
2	Primary Data Structure	--	Primary data structures are the basic data structures that directly operate upon the machine instructions.	I
3	Static Data Structure	--	A data structure formed when the number of data items are known in advance is referred as static data structure or fixed size data structure.	I
4	Static Data Structure Example	--	Arrays, Pointers, Structures	I
5	Dynamic Data Structures	--	A data structure formed when the number of data items are not known in advance is known as dynamic data structure or variable size data structure.	I
6	List some of the dynamic data structures in C	--	Linked lists, stacks, queues, trees	I
7	Linear Data Structures	--	Linear data structures are data structures having a linear relationship between its adjacent elements. E.g) Linked lists	I
8	Non-linear Data Structures	--	Non-linear data structures are data structures that don't have a linear relationship between its adjacent elements but have a hierarchical relationship between the elements. E.g) Trees and Graphs	I
9	Linked List	--	Linked list consists of a series of structures, which are not necessarily adjacent in memory.	I

10	Different types of linked lists	--	The different types of linked list include singly linked list, doubly linked list and circular linked list.	I
11	Basic operations carried out in a linked list	--	1.Creation of a list 2.Insertion of a node 3.Deletion of a node 4.Modification of a node 5.Traversal of the list	I
12	Advantages of using a linked list	--	1.It is not necessary to specify the number of elements in a linked list during its declaration. 2.Linked list can grow and shrink in size depending upon the insertion and deletion that occurs in the list.	I
13	Disadvantages of using a linked list	--	1.Searching a particular element in a list is difficult and time consuming 2. A linked list will use more storage space than an array to store the same number of elements	I
14	Applications of a linked list	--	Manipulation of polynomials, sparse matrices, stacks and queues.	I
15	Stack	--	The end from which elements are added and/or removed is referred to as top of the stack. Stacks are also referred as piles, push-down lists and last-in-first-out (LIFO) lists.	I
16	Basic operations that can be performed on a stack	--	1.Push operation 2.Pop operation 3.Peek operation 4.Empty check 5.Fully occupied check	I
17	Different ways of representing expressions	--	1.Infix Notation 2.PrefixNotation 3.Postfix Notation	I
18	Infix Notation	--	1. It is the mathematical way of representing the expression. 2. It is easier to see visually which operation is done from first to last.	I
19	Postfix Notation	--	1.Need not worry about the rules of precedence. 2. Need not worry about the rules for right to left associativity. 3. Need not need parenthesis to override the above rules.	I
20	Queue	--	Queue is an ordered collection of elements in which insertions are restricted to one end called the rear end and deletions are restricted to other end called the front end.	I

DESIGNING YOUR FUTURE
Estd. 2000

21	Priority queue	--	Priority queue is a collection of elements, each containing a key referred as the priority for that element.	I
22	Deque	--	Deque (Double-Ended Queue) is another form of a queue in which insertions and deletions are made at both the front and rear ends of the queue.	I
23	Need a Data Structure	--	A data structure helps you to understand the relationship of one data element with the other and organize it within the memory.	I
24	Abstract Data Type (ADT)	--	An abstract data type is a set of operations. ADTs are mathematical abstractions; nowhere in an ADT's definition is there any mention of how the set of operations is implemented.	I
25	Applications of stacks	--	1Towers of Hanoi 2. Reversing a string 3 Balanced parenthesis 4.Recursion using stack 5.Evaluation of arithmetic expressions	I
Unit II: Tree Structures				
26	Tree	--	A tree is a hierarchical data structure defined as a collection of nodes.	II
27	Root	--	This is the unique node in the tree to which further sub-trees are attached.	II
28	Degree of the node	--	The total number of sub-trees attached to that node is called the degree of the node.	II
29	Leaves	--	These are the terminal nodes of the tree. The nodes with degree 0 are always the leaves.	II
30	Internal nodes	--	The nodes other than the root and the leaves are called internal nodes.	II
31	Parent node	--	The node which is having further sub-branches is called the parent node of those sub-branches.	II
32	Depth and Height of a node	--	For any node ni, the depth of ni is the length of the unique path from the root to ni. The height of ni is the length of the longest path from ni to a leaf.	II
33	Depth and Height of a Tree	--	The depth of the tree is the depth of the deepest leaf. The height of the tree is equal to the height of the root. Always depth of the tree is equal to height of the tree.	II
34	Forest	--	A tree may be defined as a forest in which only a single node (root) has no predecessors. Any forest consists of a collection of trees.	II

DESIGNING YOUR FUTURE
Estd. 2000

35	Binary Tree	--	A binary tree is a finite set of nodes which is either empty or consists of a root and two disjoint binary trees called the left sub-tree and right sub-tree.	II
36	Full Binary Tree	--	A full binary tree is a tree in which all the leaves are on the same level and every non-leaf node has exactly two children.	II
37	Complete Binary Tree	--	A complete binary tree is a tree in which every non-leaf node has exactly two children not necessarily to be on the same level.	II
38	Binary Tree Traversal	--	Traversing a binary tree means moving through all the nodes in the binary tree, visiting each node in the tree only once.	II
39	Binary Tree traversal techniques	--	1. Preorder traversal 2. Inorder traversal 3. Postorder traversal 4. Levelorder traversal	II
40	In order traversal	--	1. Traverse the left sub-tree 2. Process the root node 3. Traverse the right sub-tree	II
41	Post order traversal	--	1. Traverse the left sub-tree 2. Traverse the right sub-tree 3. Process the root node	II
42	Merits of linear representation of binary trees.	--	1. Storage method is easy and can be easily implemented in arrays 2. When the location of a parent/child node is known, other one can be determined easily .	II
43	Demerit of linear representation of binary trees	--	Insertions and deletions in a node take an excessive amount of processing time due to data movement up and down the array.	II
44	Binary search tree	--	A binary search tree is a special binary tree, which is either empty or it should satisfy the following **characteristics**: Every node has a value and no two nodes should have the same value	II
45	AVL Tree	--	An empty tree is height balanced. If T is a non-empty binary tree with TL and TR as its left and right subtrees, then T is height balanced if i) TL and TR are height balanced and ii) $\lvert hL - hR \rvert \leq 1$ Where hL and hR are the heights of TL and TR respectively.	II
46	Balance factor of a node in AVL tree	--	The balance factor may be either 0 or +1 or -1. The height of an empty tree is -1.	II

47	Splay tree	--	A splay tree is a binary search tree in which restructuring is done using a scheme called splay.	II
48	Strictly binary tree	--	If every nonleaf node in a binary tree has nonempty left and right subtrees ,the tree is termed as a strictly binary tree	II
49	Different ways of representing a Binary Tree	--	i)Linear Representation using Arrays. ii)Linked Representation using Pointers	II
50	Applications of tree	--	(i) Electrical Circuit ii) Folder structure a. Binary tree is used in data processing. b. File index schemes c. Hierarchical database management system	II
Unit III : Graphs				
51	Graph	--	A Graph is a non-linear data structure consisting of nodes and edges. It can also be represented as G=(V, E).	III
52	Adjacent Nodes	--	Any two nodes which are connected by an edge in a graph are called adjacent nodes.	III
53	Un directed graph	--	A graph in which every edge is undirected is called a directed graph.	III
54	Directed graph	--	A graph in which every edge is directed is called a directed graph.	III
55	Loop	--	An edge of a graph which connects to itself is called a loop or sling.	III
56	Simple graph	--	A simple graph is a graph, which has not more than one edge between a pair of nodes than such a graph is called a simple graph.	III
57	Weighted graph	--	A graph in which weights are assigned to every edge is called a weighted graph	III
58	Out degree of a graph	--	In a directed graph, for any node v, the number of edges which have v as their initial node is called the out degree of the node v.	III
59	In degree of a graph	--	In a directed graph, for any node v, the number of edges which have v as their terminal node is called the indegree of the node v.	III
60	Simple path	--	A path in a diagram in which the edges are distinct is called a simple path. It is also called as edge simple.	III
61	Acyclic graph	--	A simple diagram which does not have any cycles is called an acyclic graph. .	III

62	Strongly connected in a graph	--	An undirected graph is connected, if there is a path from every vertex to every other vertex. A directed graph with this property is called strongly connected.	III
63	Two traversal strategies	--	a. Breadth first search b. Depth first search	III
64	Minimum spanning tree	--	A minimum spanning tree of an undirected graph G is a tree formed from graph edges that connects all the vertices of G at the lowest total cost.	III
65	Topological sort	--	A topological sort is an ordering of vertices in a directed acyclic graph, such that if there is a path from vi to vj appears after vi in the ordering.	III
66	Use of Kruskal's algorithm s	--	Kruskal's algorithm is one of the greedy techniques to solve the minimum spanning tree problem. It was discovered by Joseph Kruskal when he was a second-year graduate student	III
67	Biconnected Graph	--	A connected undirected graph is biconnected if there are no vertices whose removal disconnects the rest of the graph.	III
68	NP	--	NP is the class of decision problems for which a given proposed solution for a given input can be checked quickly to see if it is really a solution.	III
69	Shortest path problem	--	For a given graph G=(V, E), with weights assigned to the edges of G, we have to find the shortest path .	III
70	Weakly connected Graph	--	When a directed graph is not strongly connected but the underlying graph is connected, then the graph is said to be weakly connected.	III
71	Different ways of representing a graph	--	a. Adjacency matrix b. Adjacency list	III
72	Basic Operations of Graph	--	• Add Vertex − Adds a vertex to the graph. • Add Edge − Adds an edge between the two vertices of the graph.	III
73	Visiting and traversing in graph.	--	• Visiting refers to checking the value of a node when control is on the node. • Traversing means passing through nodes in a specific order.	III
74	Adjacency in graph	--	Two node or vertices are adjacent if they are connected to each other through an edge. In the following example, B is adjacent to A, C is adjacent to B, and so on.	III

DESIGNING FUTURE
Estd. 2000

75	Use of Dijksra's algorithm	--	Dijkstra's algorithm is used to solve the single-source shortest-paths problem	III
Unit IV : Introduction To Algorithms				
76	Sorting	--	Sorting arranges the numerical and alphabetical data present in a list in a specific order or sequence. There are a number of sorting techniques available.	IV
77	Types of sorting	--	• Internal sorting • External sorting	IV
78	Internal Sort	--	This is possible whenever the data to be sorted is small enough to all be held in the main memory	IV
79	External sort	--	External sorting is a term for a class of sorting algorithms that can handle massive amounts of data.	IV
80	Hashing	--	Hash function takes an identifier and computes the address of that identifier in the hash table using some function.	IV
81	Algorithm	--	An algorithm is a sequence of unambiguous instructions for solving a problem in a finite amount of time.	IV
82	Algorithm Design and Analysis of Process	--	• Understand the problem • Decide on Computational Device Exact Vs Approximate Algorithms • Algorithm Design Techniques • Design an algorithms • Prove CorrectnesS,Analyze the Algorithm • Code the Algorithm	IV
83	2 kinds of Algorithm Efficiency	--	• Time Efficiency-How fast your algorithm runs? • Space Efficiency-How much extra memory your algorithm needs?	IV
84	Specify Algorithms	--	Algorithms can be specified natural language or pseudo code.	IV
85	Pseudo Code	--	Pseudo Code is a mixture of Natural Language and Programming Language Constructs such as functions, loops, decision making statements..etc	IV
86	Important Problem Types	--	1.Sorting 2.Searching3.String Processing 4.Graph Problem5.Combinatorial Problem 6. Geometric Problem7. Numerical Problem	IV

DESIGNING YOUR FUTURE Estd. 2000

87	Sorting Problem	--	Sorting algorithm is rearrange the items of given list descending/ascending order.	IV
88	Searching Problem	--	Finding a given value, called search key given set. Searching Algorithms needs more memory space and sorted array.	IV
89	Graph Problem	--	Graph is a collection of dgs and vertices. G=(V,E). For e.g. Traversal Algorithms, Shortest Path Algorithm, Graph Coloring Problem	IV
90	Combinatorial Problem	--	This problem that ask to find combinatorial object such as permutations, combinations or a subset. Combinatorial problems are most difficult to solve. For eg Travelling sales man problem	IV
91	Features of efficient algorithm	--	• Free of ambiguity • Efficient in execution time • Concise and compact • Completeness • Definiteness Finiteness	IV
92	Big Omega Notation	--	Omega notation provides a lower bound for the function t..	IV
93	Big ‘Oh’ Notation.	--	The Big ‘Oh’ notation provides an upper bound for the function t.	IV
94	Different types of time complexity	--	• Worst case analysis • Average case analysis • Best case analysis	IV
95	Time efficiency is measured	--	Time efficiency indicates how fast an algorithm runs. Time taken by a program to complete its task depends on the number of steps in an algorithm.	IV
96	Empirical Analysis	--	It is performed by running a program implementing the algorithm on a sample of inputs and analyzing the data observed .	IV
97	Order of Algorithm	--	The order of algorithm is a standard notation of an algorithm that has been developed to represent function that bound the computing time for algorithms.	IV
98	Classify Algorithms	--	To group algorithms according to types of problem they solve .	IV
99	Brute force	--	Brute force is a straightforward approach to solving a problem, usually directly based on the problem statement and definitions of the concepts involved.	IV
100	Bubble Sort	--	Bubble Sort is a sorting algorithm used to sort list items in ascending order by comparing two adjacent values .	IV

DESIGNING YOUR FUTURE Estd. 2000

Unit V : Algorithm Design And Analysis				
101	Amortized Analysis	--	Amortized analysis finds the average running time per operation over a worst case sequence of operations .	V
102	Algorithm design techniques	--	Algorithm design techniques (or strategies or paradigms) are general approaches to solving problems algorithmatically, applicable to a variety of problems from different areas of computing.	V
103	Stepwise refinement	--	In top down design methodology the problem is solved in sequence (step by step) is known as stepwise refinement.	V
104	Characteristics of an algorithm	--	i) Input (ii)Output (iii)Definiteness (iv)Effectiveness (v) Termination	V
105	Measure input size of algorithms	--	Time taken by an algorithm grows with the size of the input. So the running time of the program depends on the size of its input.	V
106	Pseudocode	--	A pseudocode is a mixture of a natural language and programming language like constructs. A pseudocode is usually more precise than natural language.	V
107	Flowchart	--	A flowchart is a method of expressing an algorithm by a collection of connected geometric shapes containing descriptions of the algorithm's steps.	V
108	Recursive algorithm	--	An algorithm is said to be recursive if the same algorithm is invoked in the body.	V
109	Performance evaluation	--	Performance evaluation can be loosely divided into two major phases: (i) a prior estimates (performance analysis) (ii) a Posterior testing(performance measurement)	V
110	Best-case	--	The best-case step count is the minimum number of steps that can be executed for the given parameters.	V
111	Worst-case	--	The worst-case step count is the maximum number of steps that can be executed for the given parameters.	V
112	Divide and conquer	--	A divide-and-conquer algorithm recursively breaks down a problem into two or more sub-problems of the same or related type, until these become simple enough to be solved directly.	V

113	Binary search	--	Binary search is a fast search algorithm with run-time complexity of O(log n).	V
114	Greedy technique	--	Greedy technique suggests a greedy grab of the best alternative available in the hope that a sequence of locally optimal choices will yield a globally optimal solution to the entire problem.	V
115	Dynamic programming	--	Dynamic programming is an algorithm design method that can be used when a solution to the problem is viewed as the result of sequence of decisions.	V
116	Promising Node		A node in a state-space tree is said to be promising if it corresponds to a partially constructed solution that may still lead to a complete solution.	V
117	Non-Promising Node	--	A node in a state-space tree is said to be non promising if it backtracks to the node‘s parent to consider the nest possible solution for its last component.	V
118	Backtracking	--	Backtracking is a technique based on algorithm to solve problem. It uses recursive calling to find the solution by building a solution step by step increasing values with time.	V
119	Subset-Sum Problem	--	This problem find a subset of a given set S={s1,s2,……,sn} of n positive integers whose sum is equal to a given positive integer d.	V
120	Knapsack Problem	--	Given n items of known weight wi and values vi=1,2,..,n and a knapsack of capacity w, find the most valuable subset of the items that fit in the knapsack.	V
121	Branch and Bound	--	A counter-part of the backtracking search algorithm which, in the absence of a cost criteria, the algorithm traverses a spanning tree of the solution space using the breadth-first approach.	V
122	Merge sort	--	Merge sort sorts a given array A[0..n-1] by dividing it into two halves a[0..(n/2)-1] and A[n/2..n-1] sorting each of them recursively and then merging the two smaller sorted arrays into a single sorted one.	V
123	Exhaustive search	--	A brute force solution to a problem involving search for an element with a special property, usually among combinatorial objects such as permutations, combinations, or subsets of a set	V
124	Traveling Salesman Problem	--	Given a complete undirected graph G=(V, E) that has nonnegative integer cost c(u, v) associated with each edge (u, v) in E, the problem is to find a Hamiltonian cycle (tour) of G with minimum cost.	V

125	Aim of Backtracking	--	Just a path All paths the shortest path	V
Placement Questions				
126	Difference between file structure and storage structure	--	The difference is that storage structure has data stored in the memory of the computer system, whereas file structure has the data stored in the auxiliary memory.	
127	Linear data structures differ from non-linear data structures	--	Lists, stacks, and queues are examples of linear data structures whereas graphs and trees are the examples of non-linear data structures.	
128	Array	--	Arrays are the collection of similar types of data stored at contiguous memory locations.It is the simplest data structure where the data element can be accessed randomly just by using its index number	
129	Multi dimensional array	--	Multi-dimensional arrays are those data structures that span across more than one dimension.	
130	Linked list	--	A linked list is a data structure that has sequence of nodes where every node is connected to the next node by means of a reference pointer.	
131	Linked lists of linear or non-linear type	--	Linked lists can be considered both linear and non-linear data structures. This depends upon the application that they are used for When linked list is used for access strategies.	
132	Applications of queue	--	CPU Task scheduling BFS algorithm to find shortest distance between two nodes in a graph. Website request processing Used as buffers in applications like MP3 media player, CD player, etc. Managing an Input stream	
133	stack different from a queue	--	In a stack, the item that is most recently added is removed first whereas in queue, the item least recently added is removed first	
134	Hashmap in data structure	--	Hashmap is a data structure that uses implementation of hash table data structure which allows access of data in constant time (O(1)) complexity if you have the key.	

135	Priority queue	--	A priority queue is an abstract data type that is like a normal queue but has priority assigned to elements.	
136	Treedata structure Applications	--	Filesystems —files inside folders that are inturn inside other folders. Family trees — parents, grandparents, children, and grandchildren etc that represents the family hierarchy.	
137	Maximum number of nodes in a binary tree of height k	--	The maximum nodes are : 2k+1-1 where k >= 1	
138	Pre order Traversal use	--	Preorder traversal is commonly used to create a copy of the tree. It is also used to get prefix expression of an expression tree.	
139	Post order Traversal use	--	Postorder traversal is commonly used to delete the tree. It is also useful to get the postfix expression of an expression tree.	
140	Graph data structure	--	Graph is a type of non-linear data structure that consists of vertices or nodes connected by edges or links for storing data. Edges connecting the nodes may be directed or undirected.	
141	Recursive algorithm	--	Recursive algorithm is a method of solving a complicated problem by breaking a problem down into smaller and smaller sub-problems until you get the problem small enough that it can be solved easily.	
142	Time complexity of Algorithm	--	The time complexity of an algorithm denoted the total time needed by the program to run to completion. It is generally expressed by using the big O notation.	
143	Time Efficiency and Space Efficiency	--	Time Efficiency measured by estimate the number of times the essential algorithms functions are executed. Space Efficiency is measured by calculating the number of additional memory units consumed by the algorithm.	

144	Effectiveness of the algorithm	--	Input- Zero or more quantities are externally provided. Output- At least one quantity is composed Definiteness- Each instruction is simple and unambiguous Finiteness- If we trace out the instructions of an algorithm, then for all step the algorithm complete after a finite number of steps Effectiveness- Every instruction must be elementary.	
145	Linear Search	--	Linear search method is also called a sequential search technique. The linear search is a technique of searching an item in a list in sequence.	
146	Optimal solution	--	A feasible solution, which either maximizes or minimizes a given purpose method is known as an optimal solution.	
147	Warshall's Algorithm	--	Warshall's algorithm is a function of dynamic programming procedure, which is used to find the transitive closure of a directed graph.	
148	Greedy Algorithm	--	A greedy technique for an optimization problem always makes the option which look best at the moment and add it to the current sub solution.	
149	Minimum Spanning Trees	--	A spanning tree for a linked graph is a tree whose vertex set is the same as the vertex set of the given graph, and whose edge set is a subgroup of the edge set of the given graph. i.e., any linked graph will have a spanning tree.	
150	State-space tree	--	The processing of backtracking is resolved by constructing a tree of choices being made. This is known as state-space tree.	

DESIGNING YOUR FUTURE
Estd. 2000

Faculty Prepared **Signature**

Mrs.G.Krishnaveni

HoD