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

B.E - CY

2021-22

Course Code & Course Name : 19CYC01 & DATA STRUCTURES
Year/Sem/Sec : II/III/-

Subject		19CYC01 & DATA STRUCTURES		
S.No	Term	Notation (Symbol)	Concept/Definition/Meaning/Units/Equation/Expression	Units
Unit-I : Introduction and List				
1	Data		Data are simply values or sets of values	
2	Information		Processed Data	
3	Datum		Singular form of Data	
4	Data		Plural form of Data	
5	Data structures	DS	Way of organizing data in a computer called DS	
6	Classification of DS		Static data structures Dynamic data structures	
7	Static data structures		Fixed size data structure. EX: Array, pointers, structures	
8	Dynamic data structures		Variable size data structure. Ex: linked lists, stacks, queues, trees	
9	Types of data structure		Linear data structure. Non-linear data structure	
10	Linear data structures		Data are arranged in sequential order	
11	Non-linear data structure		Data structures that don't have a linear relationship between its adjacent elements but have a hierarchical relationship	
12	Abstract Data Type	ADT	Set of operations for which the implementation of the data structure is not specified	

13	Primitive data types		Each variable has a specific data typeit tells - size, range called primitive data types	
14	4 basic primitive data types		Integer , floating-point ,character and Pointer	
15	Pointer		Special type of variables that are used to store address of another variable	
16	Searching		Finding an element position in a given array called searching Type: linear search&binary search	
17	Efficiency of DS		Efficient Algorithm that takes least possible running time and consumes least memory space	
18	Asymptotic analysis		Measures the performance of the algorithm with the change in the order of the input size	
19	Case complexity		Worst case complexity, best case complexity and average case complexity	
20	Asymptotic complexity		Approximate measure of time complexity is called Asymptotic complexity	
21	Asymptotic notations		Is measured with the help of asymptotic notations	
22	Time complexity		Quantifies the amount of time taken by an algorithm to run as a function	
23	Singly linked list		Linked list elements are not stored at contiguous location	
24	Doubly linked list		Contains an extra pointer, typically called previous pointer, together with next pointer and data	
25	Circular linked list		Linked list where all nodes are connected to form a circle. There is no null at the end	
Unit-II : Stacks and Queue				
26	Array		Fixed-size DS	
27	Recursion function		Recursion is an approach in which a function calls itself with an argument	
28	Stack		Stack is an ordered collection of elements in which insertions and deletions are restricted to one end called top	
29	Top		Insertions and deletions of stack take place in top pointer	
30	Push operation		Inserting an element in stack	

31	Pop operation		Removing an element from stack	
32	Peek operation		Viewing top element of stack	
33	Empty stack		If top=-1 represent empty stack	
34	Full		If top=maxsize-1 represent full stack	
35	Queue		Queue is an ordered collection of elements in which insertions and deletions take place in 2 ends	
36	Rear end		The end from which elements are added referred to rear end	
37	Front end		End from which deletions are made is referred to as the front end	
38	Priority queue		Priority queue is a collection of elements, each containing a key referred as the priority for that element	
39	Enqueue		Inserting an element in queue	
40	Dequeue		Removing an element from queue	
41	Front		Ptr points to 1,st element of queue	
42	Rear		Ptr points to last element of queue	
43	Types of queues		• Linear queues • Circular queues • Priority queue	
44	Applications of stacks		• Reversing a string • Balanced parenthesis • Evaluation of arithmetic expressions	
45	Underflow		Checking queue is empty (contain no elements in array) called underflow	
46	Overflow		Checking queue is full (contain all elements in array) called overflow	
47	LIFO		Last in first out (principle followed by stack)	
48	FIFO		First in first out(principle followed by stack queue)	
49	Max heap		The key at root must be maximum among all keys present in binary heap	
50	Min heap		The key at root must be minimum among all keys present in binary heap	

Unit-III : Tree and Binary Search Tree				
51	Tree		A tree is a non-linear data structure, which represents hierarchical relationship between individual data items	
52	Height of a Tree		Length of the longest path from the root to a leaf	
53	Path in a tree		Sequence of distinct nodes in which successive nodes are connected by edges	
54	Leaf node		A node that has no children	
55	Binary tree nodes		A binary tree is a tree in which every non-leaf node has at most two children	
56	Full binary tree		A full binary tree is a tree in which all leaves are on the same level	
57	Complete binary tree		Is a binary tree in which every level, except possibly the last, is completely filled	
58	Right-skewed binary tree		Binary tree is a tree, which has only right child nodes	
59	Representing a binary tree		Linear representation using arrays. Linked representation using pointers.	
60	Tree traversal		Moving through all the nodes in the binary tree	
61	Types of tree traversal		• Preorder traversal • Inorder traversal • Postorder traversal	
62	Infix notation		X + Y ,Operators are written in-between their operands	
63	Postfix notation		X Y +, Operators are written after their operands.	
64	Prefix notation		+ X Y, Operators are written before their operands	
65	Other name for Postfix notation		Reverse Polish notation	
66	Other name for Prefix notation		also known as "Polish notation	
67	Post fix expression for (a+b*c)/d		abc*+d/	
68	Pre fix expression for (a+b*c)/d		/+a*bcd	
69	Head		First node of list	
70	Fields of Single linked list node		Data and next	
71	Next		Address of next node of list	

72	Fields of Double linked list node		Data, next and previous	
73	previous		Address of previous node of list	
74	Isempty of list ()		If head== NULL represent empty list	
75	Traversing		Operation perform viewing of all element in the list	
Unit-IV : Graphs				
76	Graph		A graph is a non-linear data structure that represents less relationship between its adjacent elements. There is no hierarchical relationship between the adjacent elements in case of graphs	
77	Undirected graph		If an edge between any two nodes in a graph is not directionally oriented a graph is called as undirected graph	
78	Directed graph		If an edge between any two nodes in a graph is directionally oriented, a graph is called as directed graph; it is also referred as a digraph	
79	Cycle		A cycle is a path containing at least three vertices such that the starting and the ending vertices are the same.	
80	Weighted graph		A graph is said to be weighted graph if every edge in the graph is assigned some weight or value	
81	Minimum spanning trees		A minimum spanning tree is one of the spanning trees of the graph which has the smallest sum of weights amongst all spanning trees.	
82	DFS		DFS means Depth First search it is like a preorder traversal of a tree. It is continuous searching for the unvisited nodes in the forward direction based on the recursive process	
83	Complete Graph		In a graph if there exists the path from any vertex to any other vertex, then the graph is called as Complete Graph	
84	BFS		BFS performs simultaneous exploration starting from a common point and spreading out independently	
85	Self loop		In graph theory, a loop is an edge that connects a vertex to itself	
86	Representation of Graph		• Adjacency List • Adjacency Matrix	
87	Data Structure used in BFS		Queue	

88	Data Structure used in DFS		Stack	
89	Vertex		Each node of the graph is termed as vertex	
90	Edge		Edge represents a path between two vertices	
91	Adjacency		Two nodes or vertices are adjacent if they are connected to each other	
92	Path		Path represents the series of edges between two vertices	
93	Basic operations on the graph		• Add vertex • Add Edge • Display Vertex	
94	Out Degree		Number of outgoing vertex	
95	In Degree		Number of incoming vertex	
96	Degree of a graph		Number of incident edges	
97	Cycle		Cycle is a path which starts and ends with a same vertex	
98	Connected graph		Has all pairs of vertices connected by at least one path	
99	Directed Path		It is a path of only directed edges	
100	Directed Cycle		It is a cycle of only directed edges	
Unit-V : Hashing, SearchingAndSorting				
101	Hashing		Searching technique in O(1) time complexity	
102	Hash function		Hash(key Value)=(key value) mod (Table size)	
103	Collision in hashing		When an element is inserted, it hashes to the same value as an already inserted element, and then it produces collision.	
104	Separate chaining		Separate chaining is a collision resolution technique to keep the list of all elements that hash to the same value	
105	Open addressing		Open addressing is a collision resolving strategy in which, if collision occurs alternative cells are tried until an empty cell is found	
106	Types of collision resolution strategies in open addressing		• Linear probing • Quadratic probing • Double hashing	
107	Probing		Process of getting next available hash table array cell	

108	Linear probing		F(i)=i. Hi(x)=(hash(x)+f(i))modtablesize . I=1,2,3,4…..	
109	Quadratic probing		F(i)=i 2. Hi(x)=(hash(x)+f(i))modtablesize . I=1,2,3,4…..	
110	Sorting		A sorting algorithm is used to rearrange a given array or list elements in ascending or descending order.	
111	Types of internal sorting		• Bubble Sort • Insertion Sort • Selection Sort • Quick Sort • Merge Sort • Heap Sort	
112	Classification of sorting		Internal sorting and external sorting	
113	Internal sorting		internal sorting the data that has to be sorted will be in the main memory	
114	External sorting		External sorting it will on disks, outside main memory	
115	Types of external sorting		Two-way merge sort ,radix sort	
116	Time complexity of bubble sort		Θ (n)	
117	Divide-and-Conquer		Divide: Break the given problem into sub problems of same type. Conquer: Recursively solve these sub problems Combine: Appropriately combine the answers	
118	Not a stable sorting algorithm		Bubble sort	
119	Not a stable sorting algorithm		Merge sort	
120	O(nlogn)		Running merge sort on an array of size n which is already sorted is	
121	O(n log n))		The time complexity of a quick sort algorithm	
122	Time complexity of insertion sort		Θ (n)	
123	Mod function %		Returns remainder value	
124	7%8		7	
125	10%8		2	
PLACEMENT QUESTIONS				
126	Three times the first of three consecutive		Let the three integers be x, $x + 2$ and $x + 4$. Then, $3x = 2(x + 4) + 3 \Leftrightarrow x = 11$.	

	odd integers is 3 more than twice the third. The third integer is:		∴ Third integer = $x + 4 = 15$.	
127	Look at this series: 7, 10, 8, 11, 9, 12, ...		This is a simple alternating addition and subtraction series. In the first pattern, 3 is added; in the second, 2 is subtracted.	
128	Look at this series: 22, 21, 23, 22, 24, 23, ….		In this simple alternating subtraction and addition series; 1 is subtracted, then 2 is added, and so on.	
129	Look at this series: 53, 53, 40, 40, 27, 27, ...		In this series, each number is repeated, then 13 is subtracted to arrive at the next number.	
130	Look at this series: 1.5, 2.3, 3.1, 3.9, ...		In this simple addition series, each number increases by 0.8.	
131	Three times the first of three consecutive odd integers is 3 more than twice the third. The third integer is:		Let the three integers be x, $x + 2$ and $x + 4$. Then, $3x = 2(x + 4) + 3 \Leftrightarrow x = 11$. ∴ Third integer = $x + 4 = 15$.	
132	Look at this series: 7, 10, 8, 11, 9, 12, ...		This is a simple alternating addition and subtraction series. In the first pattern, 3 isadded; in the second, 2 is subtracted.	
133	Look at this series: 22, 21, 23, 22, 24, 23, ...		In this simple alternating subtraction and addition series; 1 is subtracted, then 2 is added, and so on.	
134	(112×5^4) = ?		(112×5^4) = 112 x(10)4=112 x 10^4=1120000= 7000022^{4}16	
135	It was Sunday on Jan 1, 2006. The day of the week Jan 1, 2010 is		On 31st December, 2005 it was Saturday. Number of odd days from the year 2006 to the year 2009 = (1 + 1 + 2 + 1) = 5 days. ∴ On 31st December 2009, it was Thursday. Thus, on 1st Jan, 2010 it is Friday.	

136	Today is Monday. After 61 days, it will be:		Each day of the week is repeated after 7 days. So, after 63 days, it will be Monday. ∴ After 61 days, it will be Saturday.	
137	If 6th March, 2005 is Monday,The day of the week on 6th March, 2004 is		The year 2004 is a leap year. So, it has 2 odd days. But, Feb 2004 not included because we are calculating from March 2004 to March 2005. So it has 1 odd day only. ∴ The day on 6th March, 2005 will be 1 day beyond the day on 6th March, 2004. Given that, 6th March, 2005 is Monday. ∴ 6th March, 2004 is Sunday (1 day before to 6th March, 2005).	
138	The days in$x$ weeks x days?		x weeks x days = ($7x + x$) days = $8x$ days.	
139	On 8th Feb, 2005 it was Tuesday. The day of the week on 8th Feb, 2004 is		The year 2004 is a leap year. It has 2 odd days. ∴ The day on 8th Feb, 2004 is 2 days before the day on 8th Feb, 2005. Hence, this day is Sunday.	
140	The greatest number that will divide 43, 91 and 183 so as to leave the same remainder in each case.		Required number = H.C.F. of (91 - 43), (183 - 91) and (183 - 43) = H.C.F. of 48, 92 and 140 = 4.	
141	The H.C.F. of two numbers is 23 and the other two factors of their L.C.M. are 13 and 14. The larger of the two numbers is:		Clearly, the numbers are (23 x 13) and (23 x 14). ∴ Larger number = (23 x 14) = 322	
142	(112 x 5^4) = ?		(112 x 5^4) = 112 x(10)4=112 x 10^4=1120000= $7000022^4$16	
143	It was Sunday on Jan 1, 2006.The day of the week Jan 1, 2010 is		On 31st December, 2005 it was Saturday. Number of odd days from the year 2006 to the year 2009 = (1 + 1 + 2 + 1) = 5 days.	

			∴ On 31^{st} December 2009, it was Thursday. Thus, on 1^{st} Jan, 2010 it is Friday.	
144	Today is Monday. After 61 days, it will be:		Each day of the week is repeated after 7 days. So, after 63 days, it will be Monday. ∴ After 61 days, it will be Saturday.	
145	If 6^{th} March, 2005 is Monday, The day of the week on 6^{th} March, 2004 is		The year 2004 is a leap year. So, it has 2 odd days. But, Feb 2004 not included because we are calculating from March 2004 to March 2005. So it has 1 odd day only. ∴ The day on 6^{th} March, 2005 will be 1 day beyond the day on 6^{th} March, 2004. Given that, 6^{th} March, 2005 is Monday. ∴ 6^{th} March, 2004 is Sunday (1 day before to 6^{th} March, 2005).	
146	The daysin x weeks x days ?		x weeks x days = $(7x + x)$ days = $8x$ days.	
147	On 8^{th} Feb, 2005 it was Tuesday. The day of the week on 8^{th} Feb, 2004 is		The year 2004 is a leap year. It has 2 odd days. ∴ The day on 8^{th} Feb, 2004 is 2 days before the day on 8^{th} Feb, 2005. Hence, this day is Sunday.	
148	Find the greatest number that will divide 43, 91 and 183 so as to leave the same remainder in each case.		Required number = H.C.F. of (91 - 43), (183 - 91) and (183 - 43) = H.C.F. of 48, 92 and 140 = 4.	
149	The H.C.F. of two numbers is 23 and the other two factors of their L.C.M. are 13 and 14. The larger of the two numbers is:		Clearly, the numbers are (23 x 13) and (23 x 14). ∴ Larger number = (23 x 14) = 322	

150	Two trains running in opposite directions cross a man standing on the platform in 27 seconds and 17 seconds respectively and they cross each other in 23 seconds. The ratio of their speeds is:		Let the speeds of the two trains be x m/sec and y m/sec respectively. Then, length of the first train = 27x meters, and length of the second train = 17y meters. $\therefore \frac{27x + 17y}{x+y} = 23$ $\Rightarrow 27x + 17y = 23x + 23y$ $\Rightarrow 4x = 6y$ $\Rightarrow \frac{x}{y} = \frac{3}{2}$.	

Faculty Team Prepared **Signatures**

1. **Ms.G.Nivedhitha**

HoD