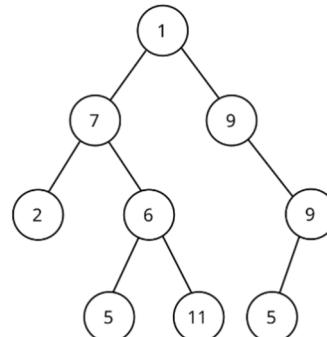


[4]

- Q.6 i. Define a graph. Explain the difference between a directed and an undirected graph and give an example of each.
- ii. Give inorder, preorder and postorder traversal of the given tree-



- OR iii. What is an AVL tree? Describe the concept of balancing in AVL trees and its importance.

4 1 2 2 1

6 3 1 3 3

Total No. of Questions: 6

Total No. of Printed Pages: 4

Enrollment No.....



Knowledge is Power

Faculty of Engineering / Science

End Sem Examination Dec 2024

CS3CO31 / BC3CO36 Data Structures

Programme: B.Tech./B.Sc.

Branch/Specialisation: CSE All/

Computer Science

Maximum Marks: 60

Duration: 3 Hrs.

Note: All questions are compulsory. Internal choices, if any, are indicated. Answers of Q.1 (MCQs) should be written in full instead of only a, b, c or d. Assume suitable data if necessary. Notations and symbols have their usual meaning.

Marks	BL	PO	CO	PSO
-------	----	----	----	-----

- Q.1 i. Which of the following supports memory addresses and direct access to elements? 1 1 2 2 1
- (a) Linked list (b) Array
 (c) Queue (d) Tree
- ii. In a multidimensional array, if the base address is 1000 and each element requires 4 bytes, what is the address of the element in the second row, 2nd column in a 2D array? Assume array of size A[2][2]. 1 1 2 2 1
- (a) 1008 (b) 1012
 (c) 1016 (d) 1020
- iii. Suppose a singly linked list is used to store integers. If the address of the first node is p, and each node has an integer field data and a pointer next, which of the following code snippets correctly reverses the list? 1 1 4 4 2
- (a) while (p != NULL) { temp = p->next; p->next = NULL; p = temp; }
 (b) while (p != NULL) { temp = p->next; p->next = prev; prev = p; p = temp; }
 (c) while (p != NULL) { prev = p; p = p->next; p->next = prev; }
 (d) while (p != NULL) { p = p->next; prev = p->next; }

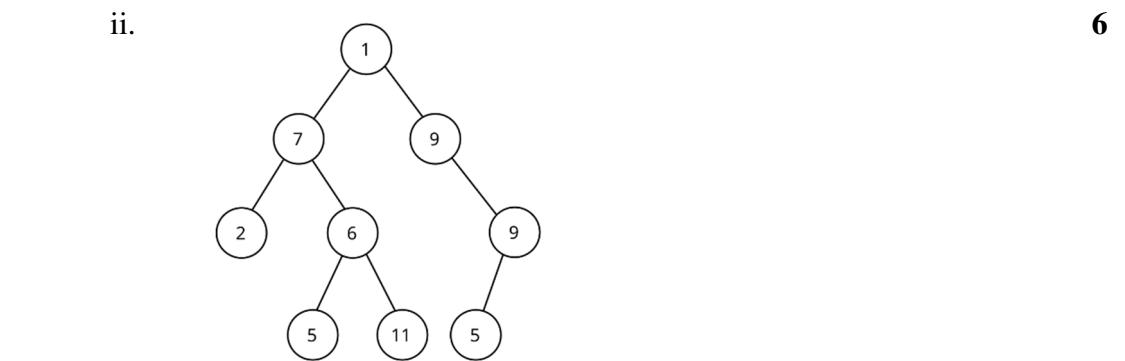
Marking Scheme
CS3CO31-BC3CO36 Data Structures

Q.1	i) (b) Array	1
	ii) (b) 1012	1
	iii) (b) while (p != NULL) { temp = p->next; p->next = prev; prev = p; p = temp; }	1
	iv) (c) Backward traversal	1
	v) (a) Overflow: top == N - 1, Underflow: top == -1	1
	vi) (c) Stack	1
	vii) (a) Bubble Sort	1
	viii) (c) Chaining	1
	ix) (c) Inorder	1
	x) (b) Complete Binary Tree	1
Q.2	i. Define data structures 2M	4
	Differentiate between linear and non-linear data structures with examples. 2M	
	ii. Write a program to reverse the content of an array in any programming language. Don't use any inbuilt library to reverse the content.	6
OR	iii. Write a program to insert an element in an array.	6
Q.3	i. Explain the structure 1M working of a doubly linked list. 1M	4
	How does it differ from a singly linked list? 2M	
	ii. insertion at the end 2M deletion from the beginning 2M display of the list. 2M	6
	OR iii. insertion at the beginning 2M deletion at the end 2M printing total number of elements in the linked 2M	6
Q.4	i. How circular queue is better than simple queue 2M Explain with example. 2M	4
	ii. Convert the following infix expression into postfix expression using stack:	6

A+B^C^D/E/F
 OR iii. Evaluate the prefix expression using stack: **6**
 $* + 5 6 - 4 2 = 22$

Q.5 i. Differentiate between linear search and binary search. **2M** **4**
 Which scenarios are best suited for each? **2M**
 ii. Explain bubble sort algorithm. **2M** **6**
 Write each and every step of sorting the following list: **4M**
 12, 18, 6, 27, 2, 16, 7
 OR iii. Explain quick sort algorithm **3M** **6**
 with proper example. **3M**

Q.6 i. Define a graph. **2M** **4**
 Explain the difference between a directed **1M**
 an undirected graph and give an example of each. **1M**



Give inorder **2M**
 preorder **2M**
 postorder traversal of the given tree. **2M**
 OR iii. What is an AVL tree **2M** **6**
 Describe the concept of balancing in AVL trees **2M**
 and its importance. **2M**

Q.4	i. How circular queue is better than simple queue 2M Explain with example. 2M	4
	ii. Convert the following infix expression into postfix expression using stack:	6