VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SLOT: B1+TB1

School of Computer Science and Engineering

FALL Semester 2024-2025

CONTINUOUS ASSESSMENT TEST-II

ANSWER KEY

Programme Name & Branch: B.Tech – Computer Science and Engineering

Course Name & code: Data Structures and Algorithms & BCSE202L

Class Number (s): COMMON TO ALL BATCHES

Faculty Name (s): All offers in B1+TB1

Exam Duration: 90 Min **Maximum Marks: 50**

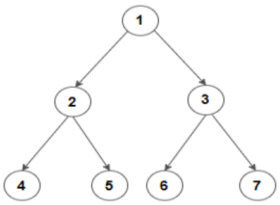
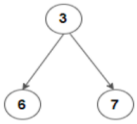
General instruction(s): Answer all the questions, Draw diagrams wherever necessary.

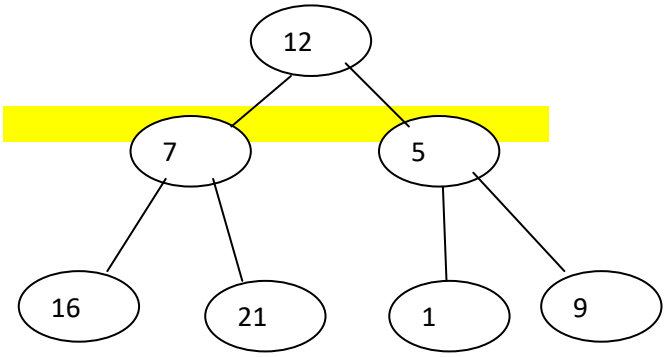
Q.No.	Question	Max Marks	CO	BL

1.	a. Write a function for reordering the elements in the single linked list as follows. Assuming the nodes are counted in a numerical fashion as **0,1,2,3, etc**, place all the even numbered nodes ahead of all odd numbered nodes. Example:
	Original List : [r,q,b,e,h,r]
	(One of the) Outputs:[r,b,h,q,e,r] **(5 Marks)**
	
	struct Node* odd_Even_List(struct Node* head)
	{
	if (!head) return head; // If the list is empty or has only one element, return
	struct Node *odd = head,
	*even = head->next,
	*even_Head = even; // Initialize odd and even pointers
	while (even && even->next)
	{
	odd->next = even->next; // Point odd nodes to the next odd node
	odd = odd->next; // Move odd pointer
	even->next = odd->next; // Point even nodes to the next even node
	even = even->next; // Move even pointer
	}
	odd->next = even_Head; // Connect the odd nodes to the even nodes
	return head; // Return the reordered list
	}
		10	CO2	BL3

	b. Write a function to check recursively the linked list of characters is palindrome or not. **(5 Marks)**
	
	// Recursive function to check if a given linked list of characters is a palindrome. Note that the left pointer is passed by reference, and the right pointer is just a copy.
	
	`isPalindrome(Node*& left, Node* right)`
	`{`
	`  // Base case`
	`  if (right == nullptr)`
	`  {`
	`  return true;`
	`  }`
	
	`  // Return false on the first mismatch`
	`  if (!isPalindrome(left, right->next))`
	`  {`
	`  return false;`
	`  }`
	
	`  // Copy the left pointer`
	`  Node* prevLeft = left;`
	
	`  // Advance the left pointer to the next node.`
	`  // This change would reflect in the parent recursive calls.`
	
	`  left = left->next;`
	
	`  // For the linked list to be a palindrome, the character at the left`
	`  // node should match with the character at the right node`
	`  return (prevLeft->data == right->data);`
	`}`			

2.	Sort the following array (Step 0 is the input) in descending order with necessary steps/diagrams and pseudocode using these properties. (i) For each input element x, the number of elements greater than x, place element x directly into its position in the output array (ii) The order of appearance of elements of the input array must be followed in the output array also if the two elements are of the same value. What is the time and space complexity of your algorithm? Step 0 : [6_1, 0_1, 2_1, 0_2, 1, 3, 4, 2_2, 6_2, 9, 5, 2_3] // actual input is [6, 0, 2, 0, 1, 3, 4, 2, 6, 9, 5, 2]. The suffix of a number tells the order of appearance if the number is appearing more than once. Eg. 6_2 tells that the number 6 is appearing 2^{nd} time. Step 1: ……	10	CO3	BL3

	…… Step n: [9,6_1,6_2,6_3,5,4,3,2_3,2_2,2_1,1,0_1,0_2] **(5+3+2=10 marks)** Working steps of Counting Sort Algorithm/Pseudocode(5 Marks) Step by step Solving(3 Marks) Time and Space Complexity(2 Marks)			
3.	a. Consider the following Binary trees Tree1 and Tree2.  Tree1  Tree2 Write a function(Pseudocode) to Determine whether "Tree2" is a subtree of another binary tree "Tree1" or not? (Note: Deploy the usage of tree traversal techniques and justify your answer). **(6 Marks)** **Idea :** i. Store inorder and postorder traversal of both trees in separate arrays. ii. Then for a given binary tree " Tree 2" to be a subset of another binary tree " Tree 1", the inorder traversal of "Tree 2" should be a subset of the inorder traversal of " Tree 1". iii. Similarly, the postorder traversal of " Tree 2" should be a subset of the postorder traversal of " Tree 1". iv. We can also perform preorder instead of the postorder traversal. For example, consider the above trees: inorder(Tree 1) = {4, 2, 5, 1, 6, 3, 7} inorder(Tree 2) = {6, 3, 7} postorder(Tree 1) = {4, 5, 2, 6, 7, 3, 1} postorder(Tree 2) = {6, 7, 3} Since inorder(Tree 2) is a subset of inorder(Tree1), and postorder(Tree 2) is a subset of postorder(Tree 1), we can say that Tree 2 is a subtree of Tree 1.	10	CO4	BL3

	b. The inorder and postorder traversal of a binary tree is given as 16, 7, 21, 12, 1, 5, 9, and 16, 21, 7, 1, 9, 5, 12. Demonstrate the construction of binary tree and give the preorder traversal. **(4 Marks)**  Pre Order Traversal : 12 7 16 21 5 1 9			

4.	a. Consider 1000 distinct elements which are stored in a binary tree. The tree is stored using the array representation of binary search tree. Assume that the array **indices** start with 0. Find out the 3rd largest element is stored at which **index**? Demonstrate your solution in pictorial representation. **(5 Marks)**	10	CO4	BL2

Answer : 509

Idea: A binary Search tree with 1000 different elements has been provided. The indices of an array begin with 0. The index number of the first node at each level may be determined by **$(2^h - 1)$** where **h** is the height of the tree. Also the number of nodes in each level may be calculated using **(2^{n-1})** where **n** is the level number.

Level	Number of nodes**(2^{n-1})**	Height	Index number of the first node at each level**$(2^h - 1)$**
1	1	0	0
2	2	1	1
3	4	2	3
4	8	3	7
5	16	4	15
6	32	5	**$(2^5 - 1)=31$**
7	64	6	**$(2^6 - 1)=63$**
8	128	7	127
9	256	8	255
10	512	9	511

At the 10^{th} level number of nodes = $(2^{10-1})= 512$

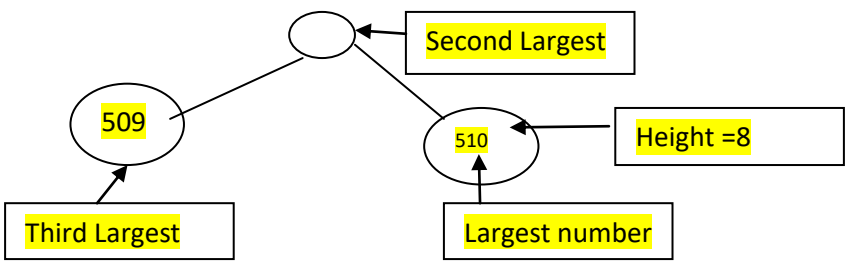
At Height 9 index number of the first node = $(2^h - 1)= (2^9 - 1)=511$

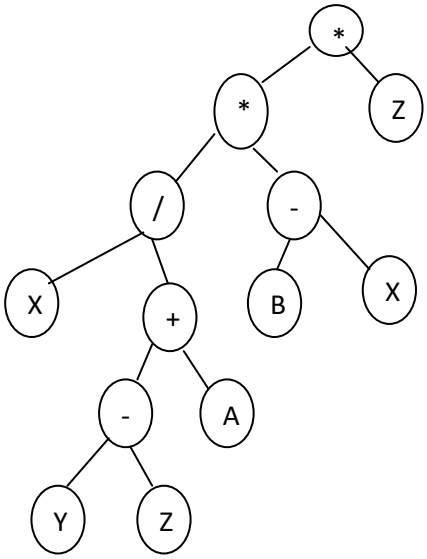
Since the total number of nodes is 1000, we need to check the upper level, because rightmost number is maximum in binary search tree.

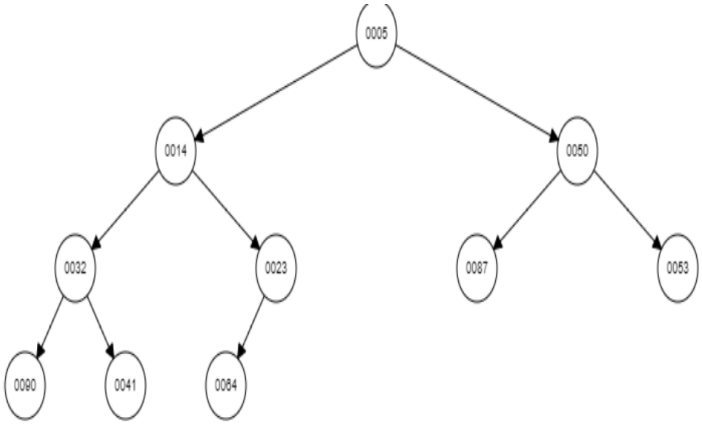
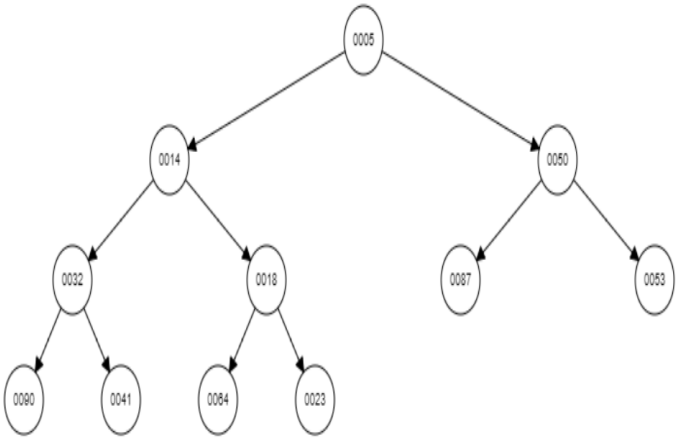
At 9^{th} level number of nodes = $(2^{9-1})= 256$

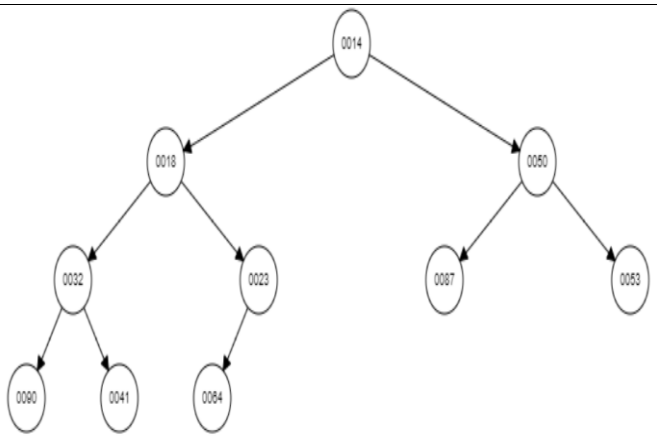
At height 8 index number of first node = $(2^8 - 1)=255$

So the index number of last node in the 9^{th} level is 255*2=510



	b. Construct the expression tree for the following expression (X/(Y-Z+A))*(B-X)*Z and give the corresponding prefix expression of the same. **(5 Marks)** 			

5.	Consider the following set of elements represented in array: [32,41,87,90,64,53,50,5,14,23] and do the following. **(10 Marks)** i. Show the construction of the tree by satisfying min heap property. (Note: Final tree is given. Based on step by step working of students, marks can be given by respective faculty)  ii. Insert 18 and show how many interchanges are required and show the new min heap tree. One interchange  iii. Delete 5 from the min-heap tree and show the step by step rearrangement.(Note: Final rearranged tree is given . Based on step by step working of students, marks can be given by respective faculty)	10	CO5	BL2

	 0014 0018 0050 0032 0023 0087 0053 0080 0041 0064			

****