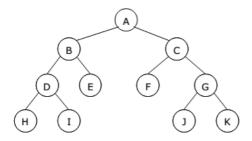|	
Started on	Friday, 16 July 2021, 4:15 PM
State	Finished
Completed on	Friday, 16 July 2021, 4:50 PM
Time taken	35 mins 1 sec

Question **1**

Complete

Marked out of 3.00

Write the pre-order traversal of the binary tree shown above.

Answer: A B D H I E C F G J K

Question **2**

Complete

Marked out of 2.50

The postorder traversal of a binary tree is 10, 8, 9, 6, 7, 4, 5, 2, 3, 1. The inorder traversal of the same tree is 10, 8, 6, 9, 4, 7, 2, 5, 1, 3. The height of the same binary tree is 5 .

Question **3**

Complete

Marked out of 2.50

The preorder traversal and inorder traversal of a binary tree is 15, 10, 12, 11, 13, 20, 18, 16, 19, 22 and 10, 11, 12, 13, 15, 16, 18, 19, 20, 22. Write the postorder traversal of the same binary tree?

Answer: 11,13,12,10,16,19,18,22,20,15

Question **4**

Complete

Marked out of 1.00

Which of the following searching technique(s) can be implemented using recursion? Choose the most appropriate option from the following.

○ Binary Search

◉ Both Linear Search and Binary Search

○ Linear Search

○ Neither Linear Search nor Binary Search

Question **5**

Complete

Marked out of 1.50

Consider an array containing elements {8, 22, 7, 9, 31, 5, 13}. The number of swapping required to sort the above mentioned array using bubble sort is 10 .

Question **6**

Complete

Marked out of 0.50

Incase of tree data structure, the children of the same parent is called as

○ descendent

○ root

◉ siblings

○ leaf node

Question **7**

Complete

Marked out of 0.50

To search for a particular element in a list which one of the following is a suitable data structure?

○ Doubly Linked List

○ Circular Linked List

◉ Array

○ None of the other options are correct

○ Singly Linked List

Question **8**

Complete

Marked out of 2.50

```
int function1(int a[], int n)
{
    int l = 0, h = n - 1;

    while (l <= h)
    {
        if (a[l] <= a[h])
        {
            return l;
        }

        int m = (l + h) / 2;

        int j = (m + 1) % n;
        int k = (m - 1 + n) % n;

        if (a[m] <= a[j] && a[m] <= a[k])
        {
            return m;
        }

        else if (a[m] <= a[h])
        {
            h = m - 1;
        }

        else if (a[m] >= a[l])
        {
            l = m + 1;
        }
    }
    return -1;
}
```

Let us assume that function1 (given above) is called from main() as "function1(a,6);" where int a[] = {18, 19, 10, 12, 15, 16}. Then the value returned by function1 is [0] .

Question **9**

Complete

Marked out of 1.00

One array contains 10 elements. After four passes, the number of elements that are placed at their final position (they will not be moved from their current location in future passes) using insertion sort is [5] .

◄ Lecture-28

Jump to...