***NAME : Himanshu Dixit***

***ENROLL NO. : 21103262***

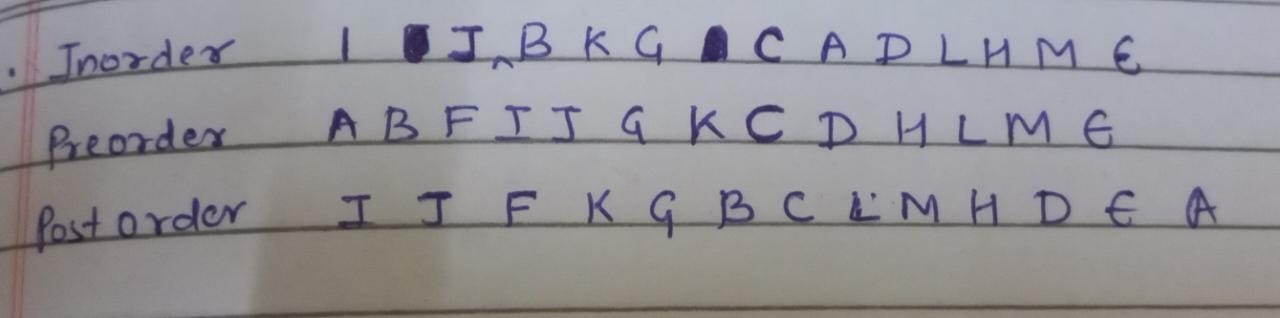
***BATCH : B11***

***Data Structure [15B11CI311]***

***Tutorial Sheet***

***Week 8***

***Answer 1)***

**

***Answer 2)***

*a) YES*

*b) A B C D E F G H I 0 0 0 0 0 0*

*c) (n-1)/2*

*d)2n+1 and 2(n+1)*

*e)efficient one as it has height of log(n)*

***Answer 3)***

******

***Answer 4)***

*a)*

*Inorder:4 2 7 5 1 3 1 0 8 6 9 11*

*Preorder:1 2 4 5 7 3 6 8 10 9 11*

*Postorder:4 7 5 10 8 11 9 6 3 1*

*d)*

*#include <bits/stdc++.h>*

*#include <iostream>*

*using namespace std;*

*class Node {*

*public:*

*int data;*

*Node \*left, \*right;*

*Node(int data)*

*{*

*this->data = data;*

*this->left = NULL;*

*this->right = NULL;*

*}*

*};*

*int findMax(Node\* root)*

*{*

*if (root == NULL)*

*return INT\_MIN;*

*int res = root->data;*

*int lres = findMax(root->left);*

*int rres = findMax(root->right);*

*if (lres > res)*

*res = lres;*

*if (rres > res)*

*res = rres;*

*return res;*

*}*

*int findMin(Node\* root)*

*{*

*if (root == NULL)*

*return INT\_MAX;*

*int res = root->data;*

*int lres = findMin(root->left);*

*int rres = findMin(root->right);*

*if (lres < res)*

*res = lres;*

*if (rres < res)*

*res = rres;*

*return res;*

*}*

*int main()*

*{*

*Node\* NewRoot = NULL;*

*Node\* root = new Node(1);*

*root->left = new Node(2);*

*root->right = new Node(3);*

*root->left->right = new Node(4);*

*root->left->right = new Node(5);*

*root->left->right->left = new Node(7);*

*root->right->right = new Node(6);*

*root->right->right->left = new Node(8);*

*root->right->right->right = new Node(9);*

*root->right->right->left->left = new Node(10);*

*root->right->right->right->right = new Node(11);*

*cout << "Maximum element is " << findMax(root) << endl;*

*cout << "Minimum element is " << findMin(root) << endl;*

*return 0;*

*}*

***Answer 5)***

*If binary tree has height h, maximum number of nodes will be when all levels*

*are completely full. Total number of nodes will be 2^0 + 2^1 + …. 2^h =*

*2^(h+1)-1.*

*If binary tree has height h, minimum number of nodes is h+1 (in case of left*

*skewed and right skewed binary* tree).