Binary search tree

Level order of tree

<https://vjudge.net/problem/HackerRank-si-level-order-of-tree/origin>

#include <iostream>

using namespace std;

struct node{

int data;

node\* left;

node\* right;

node(int value):data(value),left(nullptr),right(nullptr){}

};

node\* createNode(int value){

return new node(value);

}

node\* insert(node\* root,int value){

if(root==nullptr){

return createNode(value);

}

if(value<root->data){

root->left=insert(root->left,value);

}

else{

root->right=insert(root->right,value);

}

return root;

}

int treeheight(node\* root){

int lheight;

int rheight;

if(root==nullptr)

return 0;

else{

lheight=treeheight(root->left);

rheight=treeheight(root->right);

}

return max(lheight,rheight)+1;

}

void printcurrentlevel(node\* root,int level){

if(root==nullptr)

return;

if(level==1)

cout<<root->data<<" ";

else if(level>1){

printcurrentlevel(root->left,level-1);

printcurrentlevel(root->right,level-1);

}

}

void printlevelorder(node\* root){

int h=treeheight(root);

for(int i=1;i<=h;i++){

printcurrentlevel(root,i);

cout<<endl;

}

}

int main()

{

int T,n;

cin>>T;

while(T--){

cin>>n;

int a[n];

for(int i=0;i<n;i++){

cin>>a[i];

}

node\* root=nullptr;

for(int i=0;i<n;i++){

root=insert(root,a[i]);

}

printlevelorder(root);

cout<<endl;

}

return 0;

}

Input: 3

5

1 2 3 4 5

5

3 2 4 1 5

7

4 5 15 0 1 7 17

Output: 1

2

3

4

5

3

2 4

1 5

4

0 5

1 15

7 17