Binary search tree

Is fully binary tree

<https://vjudge.net/problem/HackerRank-si-is-full-binary-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;

}

bool isFullBinaryTree(node\* root){

if(root==nullptr)

return true;

if((root->left==nullptr&&root->right!=nullptr)||

(root->left!=nullptr&&root->right==nullptr))

return false;

return isFullBinaryTree(root->left)&&isFullBinaryTree(root->right);

}

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]);

}

if(isFullBinaryTree(root))

cout<<"True"<<endl;

else

cout<<"False"<<endl;

}

return 0;

}

Input:

3

5

1 2 3 4 5

11

8 3 30 15 40 18 12 17 25 1 7

7

4 5 15 0 1 7 17

Output:

False

True

False