1-savol

#include <iostream>

#include <vector>

using namespace std;

struct Node {

int data;

Node\* left;

Node\* right;

Node(int value) {

data = value;

left = nullptr;

right = nullptr;

}

};

Node\* constructBinaryTree(const vector<int>& elements, int start, int end) {

if (start > end)

return nullptr;

int mid = (start + end) / 2;

Node\* root = new Node(elements[mid]);

root->left = constructBinaryTree(elements, start, mid - 1);

root->right = constructBinaryTree(elements, mid + 1, end);

return root;

}

int getHeight(Node\* root) {

if (root == nullptr)

return 0;

int leftHeight = getHeight(root->left);

int rightHeight = getHeight(root->right);

return 1 + max(leftHeight, rightHeight);

}

int getLength(Node\* root) {

if (root == nullptr)

return 0;

int leftLength = getLength(root->left);

int rightLength = getLength(root->right);

return 1 + leftLength + rightLength;

}

int main() {

vector<int> elements = {47, 57, 93, 5, 30, 86, 95, 87, 34, 80, 5, 49, 43, 16, 28};

Node\* root = constructBinaryTree(elements, 0, elements.size() - 1);

int height = getHeight(root);

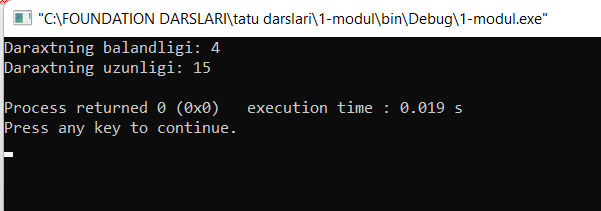
int length = getLength(root);

cout << "Daraxtning balandligi: " << height << endl;

cout << "Daraxtning uzunligi: " << length << endl;

return 0;

}



2-savol

#include <iostream>

#include <vector>

#include <queue>

using namespace std;

struct Node {

int data;

Node\* left;

Node\* right;

Node(int value) {

data = value;

left = nullptr;

right = nullptr;

}

};

Node\* constructMinHeightBST(const vector<int>& elements, int start, int end) {

if (start > end)

return nullptr;

int mid = (start + end) / 2;

Node\* root = new Node(elements[mid]);

root->left = constructMinHeightBST(elements, start, mid - 1);

root->right = constructMinHeightBST(elements, mid + 1, end);

return root;

}

void printLevelOrder(Node\* root) {

if (root == nullptr)

return;

queue<Node\*> q;

q.push(root);

while (!q.empty()) {

int levelSize = q.size();

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

Node\* current = q.front();

q.pop();

cout << current->data << " ";

if (current->left)

q.push(current->left);

if (current->right)

q.push(current->right);

}

cout << endl;

}

}

int main() {

vector<int> M = {47, 57, 93, 5, 30, 86, 95, 87, 34, 80, 5, 49, 43, 16, 28};

Node\* root = constructMinHeightBST(M, 0, M.size() - 1);

cout << "Minimal balandlikdagi daraxt massiv shaklida: " << endl;

printLevelOrder(root);

return 0;

}  
