**Height of the Binary Tree From Inorder and Level Order Traversal**

#include<bits/stdc++.h>

#include <bits/stdc++.h>

using namespace std;

class NodeInfo {

public:

int height;

int leftIndex;

int rightIndex;

NodeInfo(int h, int li, int ri) {

height = h;

leftIndex = li;

rightIndex = ri;

}

};

int heightOfTheTree(vector<int>& inorder, vector<int>& levelOrder, int N){

int maxHeight = 0;

queue<NodeInfo> q;

if (N >= 1) {

NodeInfo rootInfo(0, 0, N - 1);

q.push(rootInfo);

}

unordered\_map<int, int> indexMap;

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

indexMap[inorder[i]] = i;

}

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

NodeInfo currNodeInfo = q.front();

q.pop();

maxHeight = max(currNodeInfo.height, maxHeight);

int leftIndex = currNodeInfo.leftIndex;

int rightIndex = currNodeInfo.rightIndex;

int rootIndex = indexMap[levelOrder[i]];

if (rootIndex - 1 >= leftIndex) {

NodeInfo leftSubTree(currNodeInfo.height + 1, leftIndex, rootIndex - 1);

q.push(leftSubTree);

}

if (rootIndex + 1 <= rightIndex) {

NodeInfo rightSubTree(currNodeInfo.height + 1, rootIndex + 1, rightIndex);

q.push(rightSubTree);

}

}

return maxHeight;

}