**Construct Binary Tree From Inorder and Preorder Traversal**

TreeNode<int> \* build(vector<int> &inorder,int i\_s, int i\_e, vector<int> &preorder, int p\_s, int p\_e, map<int,int> &mp){

    if(p\_s>p\_e || i\_s>i\_e) return NULL;

    TreeNode<int> \* root = new TreeNode<int>(preorder[p\_s]);

    int index = mp[root->data];

    int numLeft = index - i\_s;

    root->left = build(inorder,i\_s,index-1,preorder,p\_s+1,p\_s+numLeft,mp);

    root->right = build(inorder,index+1,i\_e,preorder,p\_s+numLeft+1,p\_e,mp);

    return root;

}

TreeNode<int> \*buildBinaryTree(vector<int> &inorder, vector<int> &preorder)

{

    //    Write your code here

   // if(preorder.size()==0) return NULL;

    map<int,int> mp;

    for(int i=0; i<inorder.size(); i++){

        mp[inorder[i]] = i;

    }

   return build(inorder,0,inorder.size()-1,preorder, 0, preorder.size()-1,mp);

}