**Path In A Tree**

#inlcude<bits/stdc++.h>

TreeNode<int> \*buildTree(vector<int> &tree, unordered\_map<int, TreeNode<int> \*> &nodeMap, int idx)

{

    if (idx >= tree.size() || tree[idx] == -1)

        return NULL;

    if (nodeMap.find(idx) != nodeMap.end())

        return nodeMap[idx];

    TreeNode<int> \*newNode = new TreeNode<int>(tree[idx]);

    nodeMap[idx] = newNode;

    newNode->left = buildTree(tree, nodeMap, 2 \* idx + 1);

    newNode->right = buildTree(tree, nodeMap, 2 \* idx + 2);

    return newNode;

}

vector<int> pathInATree(TreeNode<int> \*root, int x)

{

   vector<int> path;

    if (root == NULL)

        return path;

    if (root->data == x)

    {

        path.push\_back(root->data);

        return path;

    }

    vector<int> leftPath = pathInATree(root->left, x);

    if (!leftPath.empty())

    {

        path.push\_back(root->data);

        path.insert(path.end(), leftPath.begin(), leftPath.end());

        return path;

    }

    vector<int> rightPath = pathInATree(root->right, x);

    if (!rightPath.empty())

    {

        path.push\_back(root->data);

        path.insert(path.end(), rightPath.begin(), rightPath.end());

        return path;

    }

    return path;

    }