**Ma’lumotlar tuzilmasi va algoritmi**

**F.I.SH. Omonbayev Jaloliddin Ravshanbek o’g’li**

**Guruh: 911-21 guruh talabasi**

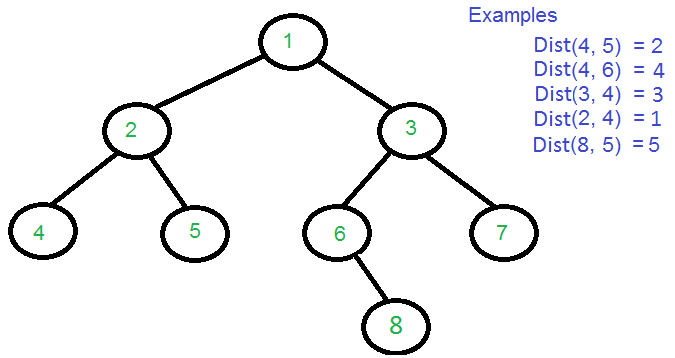
**Amaliyot: 11**

**Misol: 17 – raqam.**

**Dasturlash tili: C++ CodeBlocks**

1. Binar daraxtda berilgan tugungacha bo‟lgan masofani aniqlashning algoritmi va dasturini keltiring.

**Algaritm :**



#include <iostream>

using namespace std;

struct Node

{

struct Node \*left, \*right;

int key;

};

Node\* newNode(int key)

{

Node \*temp = new Node;

temp->key = key;

temp->left = temp->right = NULL;

return temp;

}

int findLevel(Node \*root, int k, int level)

{

if (root == NULL)

return -1;

if (root->key == k)

return level;

int l = findLevel(root->left, k, level+1);

return (l != -1)? l : findLevel(root->right, k, level+1);

}

Node \*findDistUtil(Node\* root, int n1, int n2, int &d1, int &d2, int &dist, int lvl)

{

if (root == NULL) return NULL;

if (root->key == n1)

{

d1 = lvl;

return root;

}

if (root->key == n2)

{

d2 = lvl;

return root;

}

Node \*left\_lca = findDistUtil(root->left, n1, n2, d1, d2, dist, lvl+1);

Node \*right\_lca = findDistUtil(root->right, n1, n2, d1, d2, dist, lvl+1);

if (left\_lca && right\_lca)

{

dist = d1 + d2 - 2\*lvl;

return root;

}

return (left\_lca != NULL)? left\_lca: right\_lca;

}

int findDistance(Node \*root, int n1, int n2)

{

int d1 = -1, d2 = -1, dist;

Node \*lca = findDistUtil(root, n1, n2, d1, d2, dist, 1);

if (d1 != -1 && d2 != -1)

return dist;

if (d1 != -1)

{

dist = findLevel(lca, n2, 0);

return dist;

}

if (d2 != -1)

{

dist = findLevel(lca, n1, 0);

return dist; }

return -1;

}

int main()

{

Node \* root = newNode(1);

root->left = newNode(2);

root->right = newNode(3);

root->left->left = newNode(4);

root->left->right = newNode(5);

root->right->left = newNode(6);

root->right->right = newNode(7);

root->right->left->right = newNode(8);

cout << "Dist(4, 5) = " << findDistance(root, 4, 5);

cout << "\nDist(4, 6) = " << findDistance(root, 4, 6);

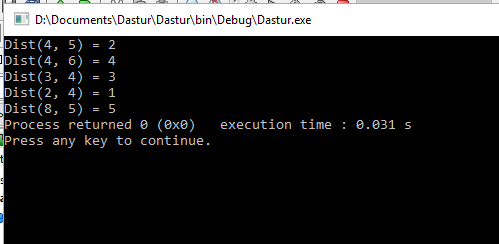
cout << "\nDist(3, 4) = " << findDistance(root, 3, 4);

cout << "\nDist(2, 4) = " << findDistance(root, 2, 4);

cout << "\nDist(8, 5) = " << findDistance(root, 8, 5);

return 0;

}



1. Bo‟sh bo‟lmagan binar daraxtning eng kattasini aniqlashning algoritmi va dasturini keltiring.

**Algaritm :**

Lightbox

#include <bits/stdc++.h>

#include <iostream>

using namespace std;

class Node {

public:

int data;

Node \*left, \*right;

Node(int data)

{

this->data = data;

this->left = NULL;

this->right = NULL;

}

};

int findMax(Node\* root)

{

if (root == NULL)

return INT\_MIN;

int res = root->data;

int lres = findMax(root->left);

int rres = findMax(root->right);

if (lres > res)

res = lres;

if (rres > res)

res = rres;

return res;

}

int main()

{

Node\* NewRoot = NULL;

Node\* root = new Node(2);

root->left = new Node(7);

root->right = new Node(5);

root->left->right = new Node(6);

root->left->right->left = new Node(1);

root->left->right->right = new Node(11);

root->right->right = new Node(9);

root->right->right->left = new Node(4);

cout << "Maximum element is " << findMax(root) << endl;

return 0;}

