BCA 613 Computer Animation I

# Ödev 4

Hazırlayan: Tayfun GÜRLEVİK

Soru: Bir n-ary ağaç yapısında Lowest Common Ancestor(LCA)’ yı bulan programı yazınız.

Çözüm:

using System;

using System.Collections.Generic;

namespace LCA\_Tree

{

class Program

{

// Maximum number of nodes is 100000 and nodes are

// numbered from 1 to 100000

static readonly int MAXN = 100001;

static List<int>[] tree = new List<int>[MAXN];

static int[,] path = new int[3, MAXN]; // storing root to node path

static bool flag;

// storing the path from root to node

static void dfs(int cur, int prev, int pathNumber, int ptr, int node)

{

for (int i = 0; i < tree[cur].Count; i++)

{

if (tree[cur][i] != prev && !flag)

{

// pushing current node into the path

path[pathNumber, ptr] = tree[cur][i];

if (tree[cur][i] == node)

{

// node found

flag = true;

// terminating the path

path[pathNumber, ptr + 1] = -1;

return;

}

dfs(tree[cur][i], cur, pathNumber, ptr + 1, node);

}

}

}

// This Function compares the path from root to 'a' & root

// to 'b' and returns LCA of a and b. Time Complexity : O(n)

static int LCA(int a, int b)

{

// trivial case

if (a == b)

return a;

// setting root to be first element in path

path[1, 0] = path[2, 0] = 1;

// calculating path from root to a

flag = false;

dfs(1, 0, 1, 1, a);

// calculating path from root to b

flag = false;

dfs(1, 0, 2, 1, b);

// runs till path 1 & path 2 mathches

int i = 0;

while (i < MAXN && path[1, i] == path[2, i])

i++;

// returns the last matching node in the paths

return path[1, i - 1];

}

static void addEdge(int a, int b)

{

tree[a].Add(b);

tree[b].Add(a);

}

// Driver code

public static void Main(String[] args)

{

for (int i = 0; i < MAXN; i++)

tree[i] = new List<int>();

// Number of nodes

addEdge(1, 2);

addEdge(1, 3);

addEdge(1, 4);

addEdge(1, 5);

addEdge(2, 6);

addEdge(2, 7);

addEdge(2, 8);

addEdge(3, 9);

addEdge(3, 10);

addEdge(4, 11);

addEdge(5, 12);

addEdge(5, 13);

addEdge(5, 14);

addEdge(6, 15);

addEdge(6, 16);

addEdge(8, 17);

addEdge(8, 18);

addEdge(8, 19);

addEdge(12, 20);

addEdge(12, 21);

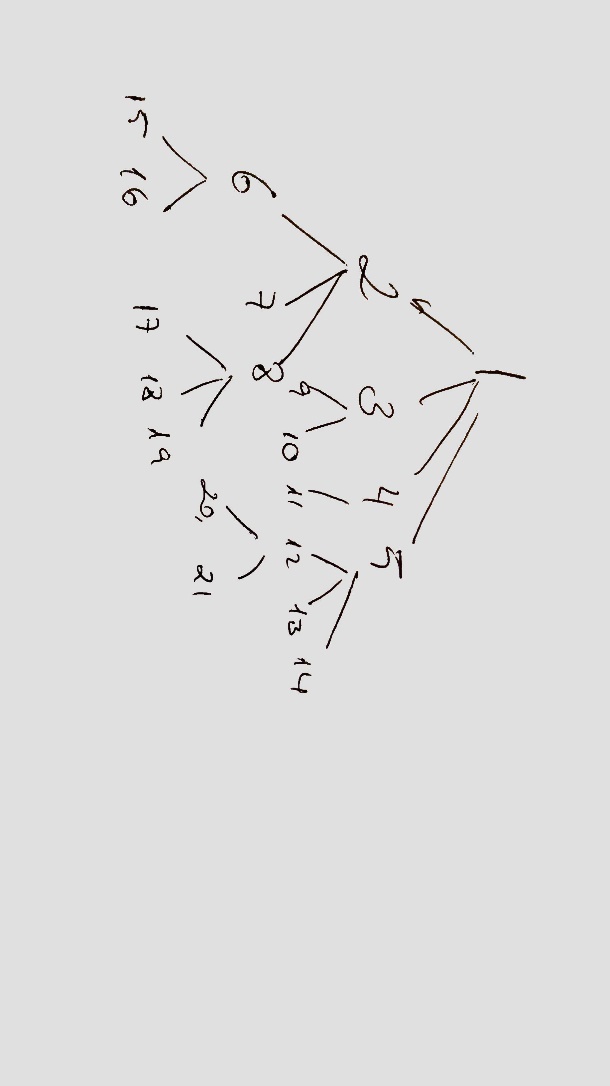
Console.Write("LCA(7, 16) = " + LCA(7, 16) + "\n");

Console.Write("LCA(14, 20) = " + LCA(14, 20) + "\n");

}

}

}



Yukarıdaki gibi bir ağaç yapısında 7 ve 16, 14 ve 20 yapraklarının en küçük ortak atalarını bulmaya çalıştığımızda alınan sonuç:

