**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информатика»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

по дисциплине **«Алгоритмы и структуры данных»**

на тему: **«**Алгоритмы на графах**»**

Выполнил: студент гр. ИП-32

Суховенко Э.С.

Принял: преподаватель Процкая М.А.

Гомель 2021

**Цель работы:** овладеть навыками алгоритмизации и программирования задач, в которых используются графы

**Вариант 12**

**Задание:**



**Исходный код программы:**

#include <iostream>

#include <vector>

#include <cmath>

using namespace std;

const int MXN = 1e4;

const int inf = 1e9;

int n, m;

bool used[MXN];

vector<vector<int>> minDist(MXN, vector<int>());

vector<int> g[MXN];

int dfs(int v, int start, int dist) {

used[v] = true;

minDist[start][v] = min(minDist[start][v], dist);

for (int x : g[v]) {

if (!used[x]) {

dfs(x, start, dist + 1);

}

}

used[v] = false;

}

int main()

{

freopen("input.txt", "r", stdin);

freopen("output.txt", "w", stdout);

cin >> n >> m;

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

int x, y;

cin >> x >> y;

g[x].push\_back(y);

g[y].push\_back(x);

}

for (int i = 1; i <= n; ++i) {

minDist[i].push\_back(0);

for (int j = 1; j <= n; ++j) {

minDist[i].push\_back(inf);

}

}

for (int i = 1; i <= n; ++i) {

dfs(i, i, 0);

}

int ans = 0;

for (int i = 1; i <= n; ++i) {

for (int j = 1; j <= n; ++j) {

ans = max(minDist[i][j], ans);

}

}

cout << ans << endl;

return 0;

}

// input.txt

10 10

1 2

1 3

3 5

4 5

4 10

5 8

6 3

6 7

7 8

9 10

// output.txt

6

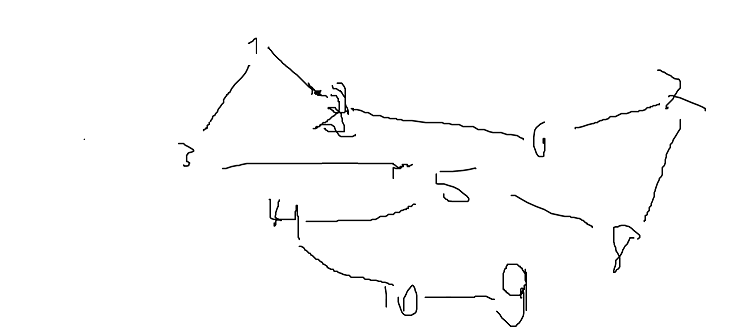


Рисунок 1 ­― граф

*Асимптотическая оценка сложности алгоритма:*

O(n^2) – посчитать наименьшую дистанцию заданной вершины.

O(n^3) – посчитать наименьшую дистанцию для всех вершин.

*Наилучшую, наихудшую и среднюю оценки:*

Все оценки равны O(n^3).

*Псевдокод операции на упрощённом языке:*

int dfs(int v, int start, int dist) {

used[v] = true;

minDist[start][v] = min(minDist[start][v], dist);

for (int x : g[v]) {

if (!used[x]) {

dfs(x, start, dist + 1);

}

}

used[v] = false;

}

for (int i = 1; i <= n; ++i) {

dfs(i, i, 0);

}

int ans = 0;

for (int i = 1; i <= n; ++i) {

for (int j = 1; j <= n; ++j) {

ans = max(minDist[i][j], ans);

}

}

**Вывод**: изучил основы работы с нелинейными динамическими структурами (бинарными деревьями).