Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №8

по курсу «Логика и основы алгоритмизации в инженерных задачах»

## на тему «Определение характеристик графа»

Выполнил:

студент группы 19ВВ2

Гусев Д. О.

Приняли:

Митрохин М. А.

Юрова О.В.

Пенза 2020

Цель работы

Реализовать алгоритм для определения характеристик графа.

Лабораторное задание

**Задание 1**

1. Сгенерируйте (используя генератор случайных чисел) матрицу

смежности для неориентированного взвешенного графа G. Выведите матрицу

на экран.

2. Определите радиус и диаметр графа G, используя матрицу смежности

графа.

3. Определите подмножества периферийных и центральных вершин

графа G, используя матрицу смежности.

4. Найдите изолированные, концевые и доминирующие вершины.

### **Листинг**

#include <time.h>

#include <iostream>

#include <stack>

#include <queue>

#include <conio.h>

using namespace std;

queue <int> Q;

int i, j, m, n, h;

int\* vis;

int\*\* graph;

int\* ex;

void BFS() {

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

printf("\nПоиск расстояний от вершины %d\n", j + 1);

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

vis[i] = 1000;

}

int s = j;

Q.push(s);

int st = s;

vis[s] = 0;

while (!Q.empty())

{

s = Q.front();

Q.pop();

for (int r = 0; r < n; r++)

if ((graph[s][r] > 0) && (vis[r] > vis[s] + graph[s][r]))

{

Q.push(r);

vis[r] = vis[s] + graph[s][r];

}

}

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

{

//вывод вершин

printf("%d ", vis[i]);

if (vis[i] > ex[st])

{

ex[st] = vis[i];

}

}

}

int rad = 100000;

int dia = 0;

printf("\nЭксцентриситеты: \n");

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

printf("%d ", ex[i]);

if (ex[i] == 0)

continue;

if (ex[i] > dia)

dia = ex[i];

if ((ex[i] < rad) && (ex[i] != -1))

rad = ex[i];

}

printf("\n\nРадиус - %d\nДиаметр - %d", rad, dia);

printf("\nЦентральные вершины: \n");

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

if (ex[i] == rad)

printf("%d ", i + 1);

}

printf("\nПериферийные вершины: \n");

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

if (ex[i] == dia)

printf("%d ", i + 1);

}

int deg = 0;

printf("\nИзолированные вершины: \n");

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

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

if (graph[i][j] != 0) {

deg += 1;

}

}

if (deg == 0) {

printf("%d ", i + 1);

}

deg = 0;

}

printf("\nДоминирующие вершины: \n");

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

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

if (graph[i][j] != 0) {

deg += 1;

}

}

if (deg == m - 1) {

printf("%d ", i + 1);

}

deg = 0;

}

printf("\nКонцевые вершины: \n");

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

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

if (graph[i][j] != 0) {

deg += 1;

}

}

if (deg == 1) {

printf("%d ", i + 1);

}

deg = 0;

}

}

void main(void)

{

setlocale(LC\_ALL, "Rus");

printf("Размерность: ");

scanf\_s("%d", &m);

printf("Диапазон числе в графе: ");

scanf\_s("%d", &h);

h++;

n = m;

graph = new int\* [m];

ex = (int\*)malloc(m \* sizeof(int));

vis = (int\*)malloc(m \* sizeof(int));

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

graph[i] = new int[m];

}

//генерация матрицы

printf("Матрица смежности: \n");

srand(time(NULL));

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

for (j = i; j < m; ++j) {

graph[i][j] = graph[j][i] = rand() % h;

}

graph[i][i] = 0; //обнуление вершины

}

h--;

printf("\n");

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

{

for (j = 0; j < m; ++j)

printf("%d ", graph[i][j]);

printf("\n");

}

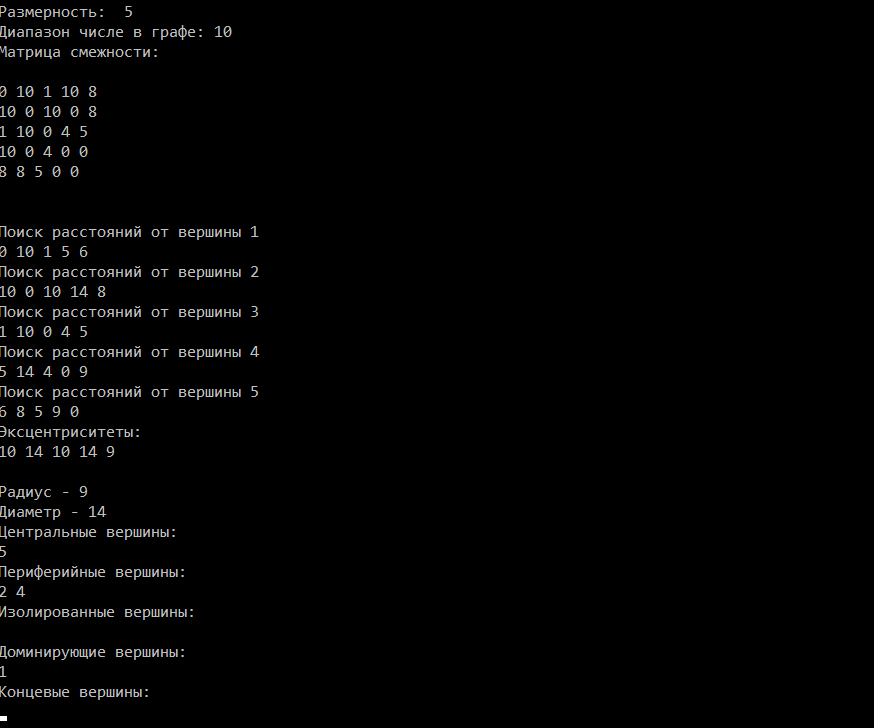
printf("\n");

BFS();

\_getch();

}

**Результаты работы программы**



## Вывод

Научились реализовывать алгоритм поиска радиуса и диаметра графа и прочих его характеристик.