Министерство науки и образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

ОТЧЕТ

по лабораторной работе №10

по курсу «Логика и основы алгоритмизации в инженерных задачах»

Выполнили:

студенты группы 21ВВ2

Сорокина Е.А.

Нефедова Е.Д.

Принял:

Митрохин М.А.

Юрова О.В.

Пенза 2022

**Цель:** научиться определять эксцентриситеты, диаметры, радиусы, периферийные и центральные вершины.

**Задание 1**

1. Сгенерируйте (используя генератор случайных чисел) матрицу

смежности для неориентированного взвешенного графа G. Выведите матрицу

на экран.

2. Для сгенерированного графа осуществите процедуру поиска

расстояний, реализованную в соответствии с приведенным выше описанием.

При реализации алгоритма в качестве очереди используйте класс queue из

стандартной библиотеки С++.

**Задание 2**

1. Для каждого из вариантов сгенерированных графов (ориентированного

и не ориентированного) определите радиус и диаметр.

2. Определите подмножества периферийных и центральных вершин.

**Листинг**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <malloc.h>

#include <stdlib.h>

#include <time.h>

#include <conio.h>

#include <iostream>

#include <queue>

using namespace std;

int\*\* a;

int\* d;

int i, j, n, h;

int\* vis;

queue <int> q;

void BFS(int s)

{

h = s;

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

{

vis[i] = -1;

}

q.push(s);

vis[s] = 0;

while (!q.empty())

{

s = q.front();

//printf(" %d", s);

q.pop();

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

{

if (a[s][i] > 0 && vis[i] == -1)

{

q.push(i);

vis[i] = vis[s] + a[s][i];

}

}

}

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

{

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

}

for (int l = 0; l < n; l++) {

if (vis[l] > d[h]) {

d[h] = vis[l];

}

}

printf("\n");

printf("Эксцентриситет: %d", d[h]);

printf("\n");

}

int main()

{

int dm = 0;

int rd = 1000;

system("chcp 1251");

system("cls");

printf("Введите количество вершин: ");

scanf("%d", &n);

a = (int\*\*)malloc(n \* sizeof(int\*));

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

d = (int\*)malloc(n \* sizeof(int));

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

{

d[i] = 0;

}

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

{

vis[i] = -1;

}

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

{

a[i] = (int\*)malloc(n \* sizeof(int));

}

srand(time(NULL));

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

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

{

if (i != j) {

a[i][j] = rand() % 10;

a[j][i] = a[i][j];

}

else a[i][j] = 0;

}

printf("M1:\n");

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

{

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

printf("%3d", a[i][j]);

printf("\n");

};

printf("\n");

printf("\n");

for (int k = 0; k < n; k++) {

BFS(k);

//std::cout << "1 ";

}

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

{

printf("%3d", d[j]);

}

printf("\n");

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

if (d[i] > dm) {

dm = d[i];

}

if (d[i] < rd) {

rd = d[i];

}

}

printf("\n");

printf("Диаметр: %d", dm);

printf("\n");

printf("Радиус: %d", rd);

printf("\n");

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

if (d[i] == dm) {

printf("Периферийная вершина: %d", i);

printf("\n");

}

if (d[i] == rd) {

printf("Центральная вершина: %d", i);

printf("\n");

}

}

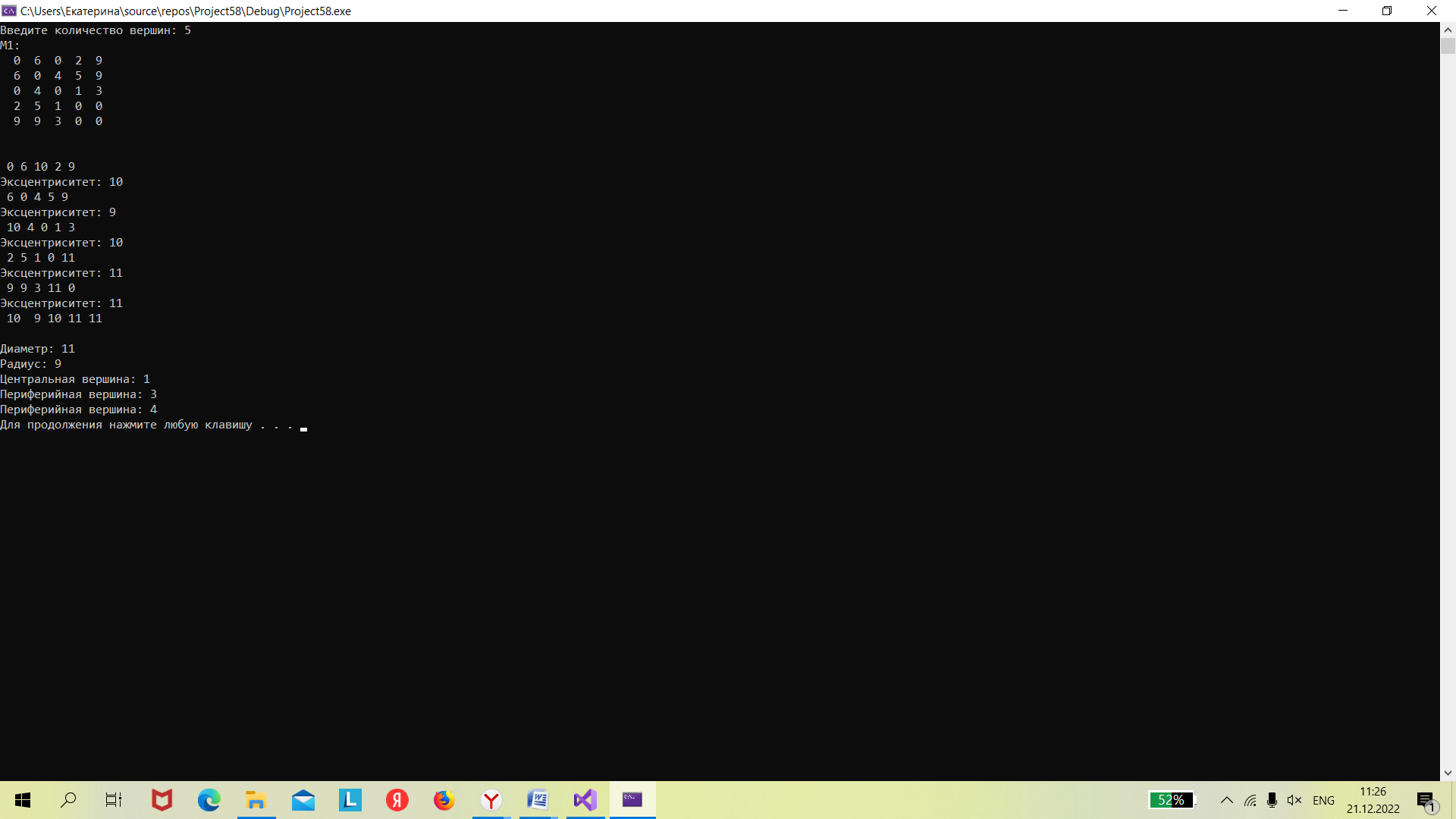
delete[]vis;

free(a);

system("pause");

}

**Результат работы программы:**



**Вывод:** в ходе данной лабораторной работы мы научились определять эксцентриситеты, диаметры, радиусы, периферийные и центральные вершины.