Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №10

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Поиск расстояний в графе»

**Выполнили студенты группы 21вв1:**

Вартанов Алексей

**Приняли**

Юрова О.В.

Акифьев И.В.

Пенза 2022

**Методическое указание:**

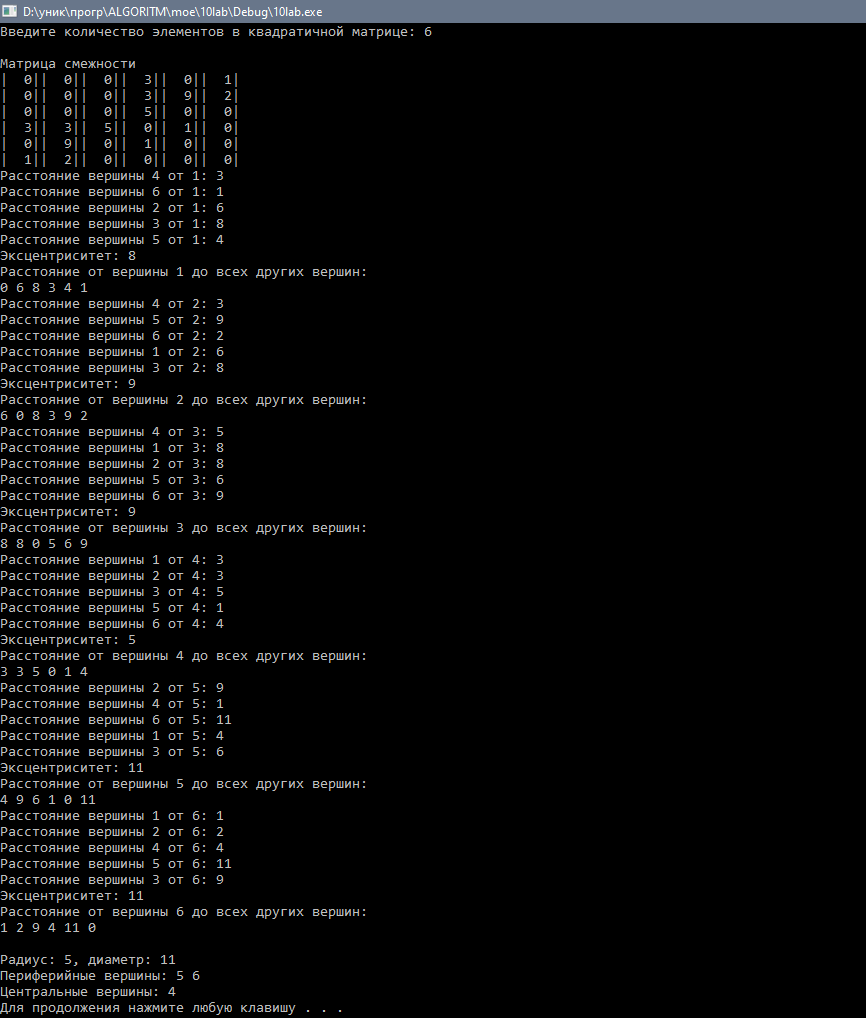
**Задание 1**

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.

**Результат программы:**

****

**Листинг:**

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <locale.h>

#include <Windows.h>

#include <malloc.h>

#include <time.h>

#include <iostream>

#include <queue>

using namespace std;

int size;

int\* vis;

int v = 0;

int\*\* M;

int\* maximum;

void DFS(int s) {

queue <int> q;

q.push(s-1);

vis[s-1] = 1;

int st = s;

while (!q.empty()) {

s = q.front();

q.pop();

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

if ((M[s][i] != 0) && (vis[i] < 0)) {

q.push(i);

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

printf("Расстояние вершины %d от %d: %d\n", i+1, st, vis[i]-1);

}

}

}

maximum[st-1]=0;

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

if (maximum[st-1]<vis[i]){

maximum[st-1]=vis[i]-1;

}

}

printf("Эксцентриситет: %d\n", maximum[st-1]);

printf("Расстояние от вершины %d до всех других вершин:\n", st);

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

printf("%d ", vis[i]-1);

}

printf("\n");

}

void main() {

setlocale(LC\_ALL, "rus");

printf("Введите количество элементов в квадратичной матрице: ");

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

srand((unsigned)time(NULL));

M = (int\*\*)malloc(sizeof(int\*) \* size);

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

M[i] = (int\*)malloc(sizeof(int) \* size);

}

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

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

vis[i] = -1;

}

maximum = (int\*)malloc(sizeof(int) \* size);

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

maximum[i] = 0;

}

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

for (int j = i; j < size; j++) {

if (rand() % 2) {

M[i][j] = rand() % 10;}

else {

M[i][j] = 0;

}

if (i == j) {

M[i][j] = 0;

}

M[j][i] = M[i][j];

}

}

printf\_s("\nМатрица смежности");

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

printf\_s("\n");

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

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

}

}

printf("\n");

for (int s = 1; s < (size + 1); s++) {

DFS(s);

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

vis[i] = -1;

}

}

printf("\n");

int maxm=0;

int minm=100;

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

if (maxm<maximum[i]){

maxm=maximum[i];

} else if (minm>maximum[i] && maximum[i]>0){

minm=maximum[i];

}

}

printf("Радиус: %d, диаметр: %d\n", minm, maxm);

printf("Периферийные вершины: ");

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

if (maximum[i]==maxm){

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

}

}

printf("\n");

printf("Центральные вершины: ");

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

if (maximum[i]==minm){

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

}

}

printf("\n");

system("pause");

}

**Вывод:**

В ходе выполнения лабораторной работы были разработана программа, выполняющая поиск расстояний в графе.

Я получил опыт в создании проектов в среде Microsoft Visual Studio, научился писать и отлаживать программы с применением поиска расстояний на языке Си.