Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №10

по дисциплине: «Логика и основы алгоритмизации в инженерных задачах.»

на тему: «Поиск расстояний во взвешенном графе.»

Выполнили:

студенты группы 21ВВ4

Колокольцева У. А.

Нагорная Д. А.

Принял:

Акифьев И. В.  
Юрова О. В.

Пенза, 2022

**Цель работы:** разработать программный код для поиска расстояний во взвешенном графе.

**Лабораторная работа:**

**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:**

#include <stdio.h>

#include <conio.h>

#include <Windows.h>

#include <stdlib.h>

#include <queue>

#include <iostream>

using namespace std;

int BFS(int\*\* A, int\* M, int i, int N) {

queue <int> Q;

int j;

Q.push(i);

M[i] = 0;

while (Q.empty() != true) {

i = Q.front();

Q.pop();

for (j = 0; j < N; j++) {

if ((A[i][j] != 0) && (M[j] > M[i] + A[i][j])) {

M[j] = 0;

Q.push(j);

M[j] = M[i] + A[i][j];

}

}

}

return 0;

}

int main(void)

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int N, i, j, C = 0;

printf("Ведите размер квадратной матрицы: ");

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

srand(time(NULL));

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

int\* EX = (int\*)malloc(N \* sizeof(int));

int\*\* A = (int\*\*)malloc(N \* sizeof(int\*));

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

A[i] = (int\*)malloc(N \* sizeof(int));

}

printf("\n Матрица смежности для взвешенного неориентированного графа \n\n");

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

for (j = 0; j < N; j++) {

A[i][j] = rand() % 10 + 1;

if (A[i][j] > 5) {

A[i][j] = 0;

}

A[j][i] = A[i][j];

if (i == j) {

A[i][j] = 0;

}

}

}

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

for (j = 0; j < N; j++) {

printf("%2d ", A[i][j]);

}

printf("\n");

}

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

M[i] = 1000;

}

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

EX[i] = -1;

}

printf("\n\nПоиск эксцентриситетов :\n");

int D = -1, r = 10000;

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

BFS(A, M, i, N);

printf("\n");

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

printf("%3d ", M[l]);

}

int k;

for (k = 0; k < N; k++) {

if ((EX[i] < M[k]) && (M[k] != 1000)) {

EX[i] = M[k];

}

}

if ((EX[i] < r) && (EX[i] != 0)) {

r = EX[i];

}

if (EX[i] > D) {

D = EX[i];

}

printf(" Эксцентириситет вершины %d : %d\n", i + 1, EX[i]);

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

M[l] = 1000;

}

}

printf("\n\nРадиус : %d\n", r);

printf("Диаметр : %d\n", D);

printf("Центральные вершины: ");

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

if (EX[n] == r) {

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

}

}

printf("\nПериферийные вершины: ");

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

if (EX[n] == D) {

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

}

}

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

int step = 0;

for (j = 0; j < N; j++) {

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

step++;

}

}

}

free(A);

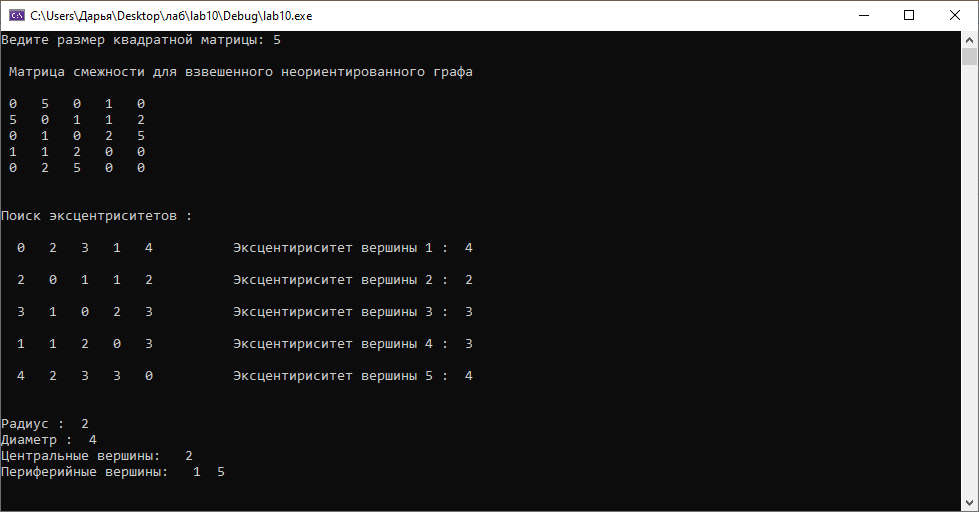
free(M);

free(EX);

\_getch();

}

Результат работы программы:



**Вывод:** в ходе данной лабораторной работы был разработан программный код для поиска расстояний во взвешанном графе, используя обход графа в ширину.