Министерство 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 графов»

Выполнили:

студенты группы 19ВВ4

Каравайкин И.М.

Савельев А.А.

Приняли:

Митрохин М.А.

Юрова О.В.

Пенза 2020

### Цель работы

Определить характеристики 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.

### Описание решения

Диаметр это максимальное расстояние до наиболее удаленной вершины в графе. Чтобы его найти нужно проверить длины путей от всех вершин в графе.

Радиус это минимальное расстояние среди эксцентриситетов (наибольших расстояний до наиболее удаленной вершины графа) Чтобы его найти, нужно также проверить длины путей от всех вершин и найти там минимальный эксцентриситет.

Множества радиусов это количество вершин, в которых эксцентриситет равен радиусу.

Множество диаметров это количество вершин, в которых эксцентриситет равен диаметру.

Изолированные вершины – вершины у которых нет смежных вершин

Концевые вершины – вершины у которых есть одна смежная вершина

Доминирующая вершина – вершина смежная со всеми другими вершинами

### Листинг

#define \_CRT\_SECURE\_NO\_WARNINGS

#define \_CRT\_SECURE\_NO\_WARNING

#include "stdio.h"

#include <stdlib.h>

#include "conio.h"

#include "malloc.h"

#include <locale>

#include <queue>

using namespace std;

int diametr = -1;

int Rad = -1;

int radius = 20;

void bfsd(int v, int\*\* arr, int\* dist, int m) {

queue <int> q;

q.push(v);

dist[v] = 0;

printf("Порядок обхода: ");

while (!q.empty()) {

v = q.front();

q.pop();

printf("%d, ", v);

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

if ((arr[v][j] != 0) && (dist[j] == -1)) {

q.push(j);

dist[j] = dist[v] + arr[v][j];

if (diametr < dist[j]) { diametr = dist[j]; }

if (Rad < dist[j]) { Rad = dist[j]; }

}

}

}

}

void main() {

srand(time(NULL));

setlocale(LC\_ALL, "Russian");

int n;

printf("Введите размер массива: ");

scanf("%d", &n);

int\*\* arr1 = (int\*\*)malloc(n \* sizeof(int\*));

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

arr1[i] = (int\*)malloc(n \* sizeof(int));

}

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

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

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

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

}

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

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

arr1[i][j] = rand() % 2;

if (arr1[i][j] == 1) {

arr1[i][j] = rand() % 5;

}

}

}

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

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

arr1[j][i] = arr1[i][j];

if (i == j) { arr1[i][j] = 0; }

printf("%d ", (arr1[i][j]));

}

printf("\n");

}

printf("\n\n\n");

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

dista[i] = -1;

}

for (int i = 0, j = 0; i < n; i++) {

bfsd(i, arr1, dista, n);

printf("\nДлина путей: ");

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

distaMn[j][i] = dista[i];

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

dista[i] = -1;

if ((Rad <= diametr) && (Rad <= radius) && (Rad != -1)) { radius = Rad; }

}

printf("\n");

j++;

if ((i == 0) && (Rad != -1)) { radius = Rad; }

Rad = -1;

printf("\n");

}

printf("\n\n\n");

if (radius == 20) { radius = diametr; }

printf("Диаметр графа - %d ", diametr);

printf("Радиус графа - %d ", radius);

printf("\n"); printf("\n"); printf("\n");

Rad = -1;

printf("Множество радиусов графа:\n");

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

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

if (Rad < distaMn[i][j]) {

Rad = distaMn[i][j];

}

}

if (Rad == radius) {

printf("%d, ", i);

}

Rad = -1;

}

printf("\n");

printf("Множество диаметров графа:\n");

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

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

if (Rad < distaMn[i][j]) {

Rad = distaMn[i][j];

}

}

if (Rad == diametr) {

printf("%d, ", i);

}

Rad = -1;

}

printf("\n");

printf("\n");

printf("Изолированные вершины:\n");

int ch = 0;

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

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

if (arr1[i][j] > 0) { ch = ch + 1; }

}

if (ch == 0) { printf("%d ", i); }

ch = 0;

}

printf("\n");

printf("Концевые вершины:\n");

ch = 0;

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

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

if (arr1[i][j] > 0) { ch = ch + 1; }

}

if (ch == 1) { printf("%d ", i); }

ch = 0;

}

printf("\n");

printf("Доминирующие вершины:\n");

ch = 0;

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

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

if (arr1[i][j] > 0) { ch = ch + 1; }

}

if (ch == (n - 1)) { printf("%d ", i); }

ch = 0;

}

}

### Результат работы

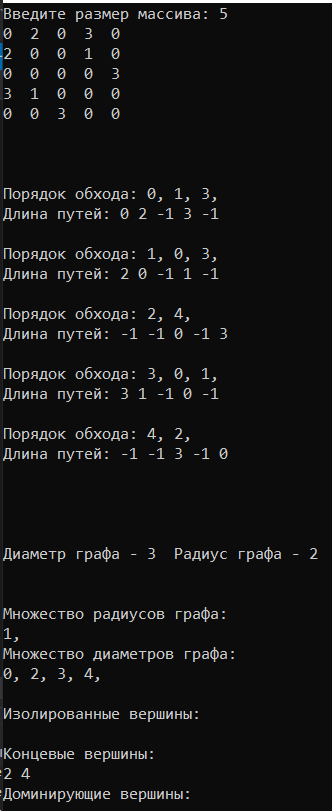


Рисунок 1 – Результат работы

### Выводы

В ходе выполнения лабораторной работы была разработана программа, определяющая характеристики графа.