|  |
| --- |
| Министерство образования Российской Федерации  Пензенский государственный университет  Кафедра «Вычислительная техника» |
| Отчет  по лабораторной работе №8  по курсу «Л и ОА в ИЗ» на тему «Определение характеристик графов» |
|  |
|  |
| Выполнил:  студент группы 20ВВ4  Кривцов Н.А.  Слабинский Н.А.  Приняли:  Юрова О.В.  Акифьев И.В. |
| Пенза 2021 |

# **Цель работы:** определить характеристики графов.

**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.

**Теоретическая часть:**

Эксцентриситет вершины – расстояние до наиболее удаленной вершины графа.

Максимальный эксцентриситет среди эксцентриситетов всех вершин графа называется диаметром графа G и обозначается через D(G).

Вершина v i называется периферийной, если её эксцентриситет равен диаметру графа.

Минимальный из эксцентриситетов вершин графа называется его радиусом и обозначается через r(G).

Вершина v i называется центральной, если её эксцентриситет равен радиусу графа.

Множество всех центральных вершин графа называется его центром.

Граф G может иметь единственную центральную вершину или несколько центральных вершин.

Степенью вершины графа G называется число инцидентных ей ребер. Степень вершины v i обозначается через deg (v, i ).

Вершина v i со степенью 0 называется изолированной, со степенью 1 – концевой.

Вершина графа, смежная с каждой другой его вершиной, называется доминирующей.

**Практическая часть:**

**Листинг:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "locale"

#include "stdio.h"

#include "conio.h"

#include <locale.h>

#include <stdlib.h>

#include <malloc.h>

#include <queue>

using namespace std;

queue <int> q;

void BFSD(int v, int\*\* G, int\* DIST, int size) {

q.push(v);

DIST[v] = 0;

while (!q.empty()) {

v = q.front();

q.pop();

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

if (G[v][i] > 0 and DIST[i] > (DIST[v] + G[v][i])) {

q.push(i);

DIST[i] = DIST[v] + G[v][i];

}

}

}

}

int main()

{

srand(time(NULL));

setlocale(LC\_ALL, "RUS");

int v, n;

int k = 0;

printf("Введите размер матрицы: ");

scanf("%d", &n);

int\*\* M;

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

int\* A;

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

int\* E;

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

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

{

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

}

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

{

A[i] = 10000;

}

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

{

E[i] = i;

}

printf("Матрицы:");

printf("\n");

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

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

if (j > k) {

int random = rand() % 2;

if (random == 0) {

M[i][j] = 0;

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

}

else {

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

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

}

}

else if (j < k) {

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

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

}

else {

M[i][j] = 0;

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

}

}

k++;

printf("\n");

} // Создание матрицы

printf("\n\n");

printf("\n");

int rad = 10000, dim = 0, max;

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

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

{

A[i] = 10000;

}

BFSD(j, M, A, n);

max = 0;

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

if (A[i] > max) max = A[i];

printf("Расстояние от вершины %d до вершины №%d = %d\n", j, i, A[i]);

}

E[j] = max;

printf("эксцентриситет = %d\n", max);

if (dim < max) dim = max;

if (rad > max and rad != 0) rad = max;

printf("\n\n");

}

printf("Диаметр графа = %d\n", dim);

printf("Радиус графа = %d\n", rad);

printf("\n\n");

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

if (E[i] == rad) printf("Вершина №%d является центральной\n", i);

}

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

if (E[i] == dim) printf("Вершина №%d является переферийной\n", i);

}

printf("\n\n");

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

int count = 0;

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

if (M[i][j] != 0) count++;

}

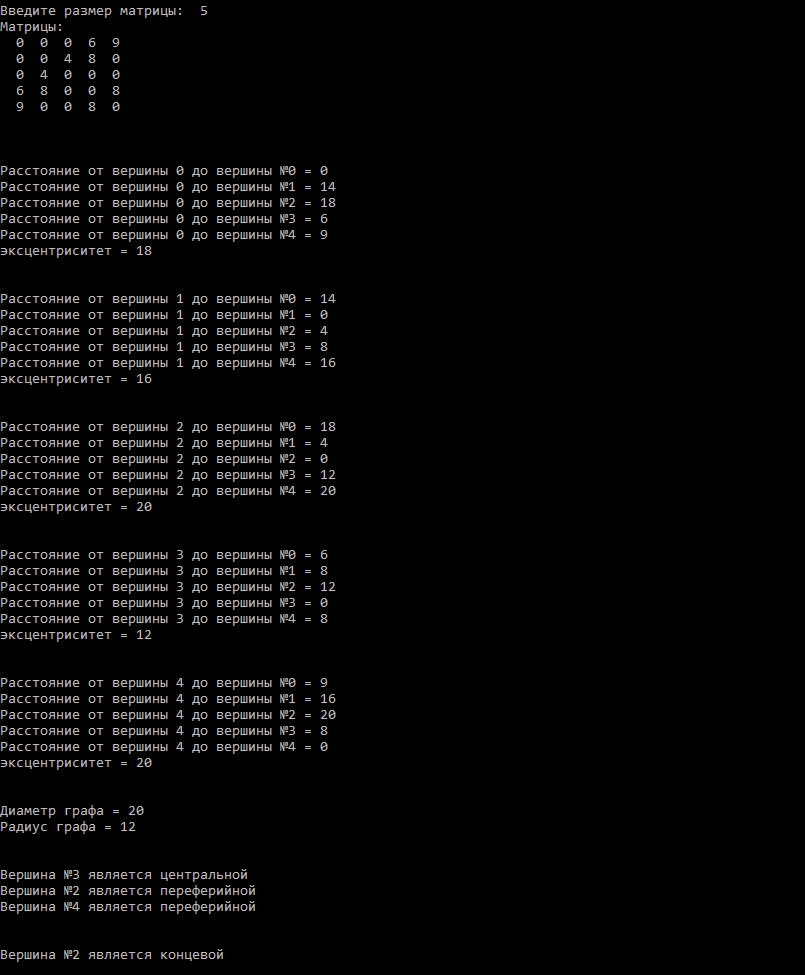
if (count == 0) printf("Вершина №%d является изолированной\n", i);

if (count == 1) printf("Вершина №%d является концевой\n", i);

if (count == n - 1) printf("Вершина №%d является доминирующей\n", i);

}

}



**Вывод:** научились определять характеристики графов