Пензенский государственный университет  
Кафедра «Вычислительной техники»

**Отчет**по лабораторной работе №8  
по дисциплине: «Логика и основы алгоритмизации в инженерных задачах»  
на тему: «Определение характеристик графов»

**Выполнил студент группы 19ВВ1:**

Яцков В.В

**Приняли:**

д.т.н. профессор Митрохин М. А.

Пенза 2020.

**Цель работы:** научится находить эксцентриситет вершины, определять радиус и диаметр графа, находить изолированные, концевые и доминирующие вершины.

**Листинг:**

#include <time.h>

#include <iostream>

#include <stack>

#include <queue>

#include <conio.h>

using namespace std;

queue <int> Q;

int i, j, m, n, h;

int\* vis;

int\* v;

int\*\* graph;

int\* ex;

int\* e;

void BFS() {

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

printf("\nПоиск расстояний от вершины %d\n", j + 1);

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

vis[i] = 1000;

v[j] = 0;

}

int s = j;

Q.push(s);

int st = s;

vis[s] = 0;

while (!Q.empty())

{

s = Q.front();

Q.pop();

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

if ((graph[s][r] > 0) && (vis[r] > vis[s] + graph[s][r]))

{

Q.push(r);

vis[r] = vis[s] + graph[s][r];

}

}

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

{

//проверка на изолированную вершину

if (vis[i] == 1000) {

printf("- ");

vis[i] = 0;

}

//вывод вершин

else

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

v[j] += vis[i];

if (vis[i] > ex[st])

{

ex[st] = vis[i];

}

}

}

int k = 0;

int rad = 100000;

int dia = 0;

printf("\nЭксцентриситеты: \n");

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

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

if (ex[i] == 0)

continue;

if (ex[i] > dia)

dia = ex[i];

if ((ex[i] < rad) && (ex[i] != -1))

rad = ex[i];

}

printf("\n\nРадиус - %d\nДиаметр - %d", rad, dia);

printf("\nЦентральные вершины: \n");

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

if (ex[i] == rad)

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

}

printf("\nПериферийные вершины: \n");

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

if (ex[i] == dia)

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

}

printf("\nИзолированные вершины: \n");

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

if (ex[i] == 0) {

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

}

}

printf("\nДоминирующие вершины: \n");

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

if (e[i] == m - 1)

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

}

printf("\nКонцевые вершины: \n");

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

if (e[i] == 1)

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

}

}

void DFS\_main()

{

setlocale(LC\_ALL, "Rus");

printf("Размерность: ");

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

printf("Максимальный вес ребра: ");

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

h++;

n = m;

int p = 0;

graph = new int\* [m];

v = (int\*)malloc(m \* sizeof(int));

ex = (int\*)malloc(m \* sizeof(int));

e = (int\*)malloc(m \* sizeof(int));

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

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

graph[i] = new int[m];

}

//генерация матрицы

int start;

printf("Матрица смежности: \n");

srand(time(NULL));

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

for (j = i; j < m; ++j) {

graph[i][j] = graph[j][i] = rand() % h;

}

graph[i][i] = 0; //обнуление вершины

}

//нумерование столбцов

printf(" ");

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

{

printf("k%d ", j + 1);

}

printf("\n");

//нумерование строк

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

{

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

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

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

printf("\n\n");

}

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

e[i] = 0;

}

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

{

printf("\n%d - ", i + 1);

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

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

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

e[i]++;

}

}

}

printf("\n");

BFS();

\_getch();

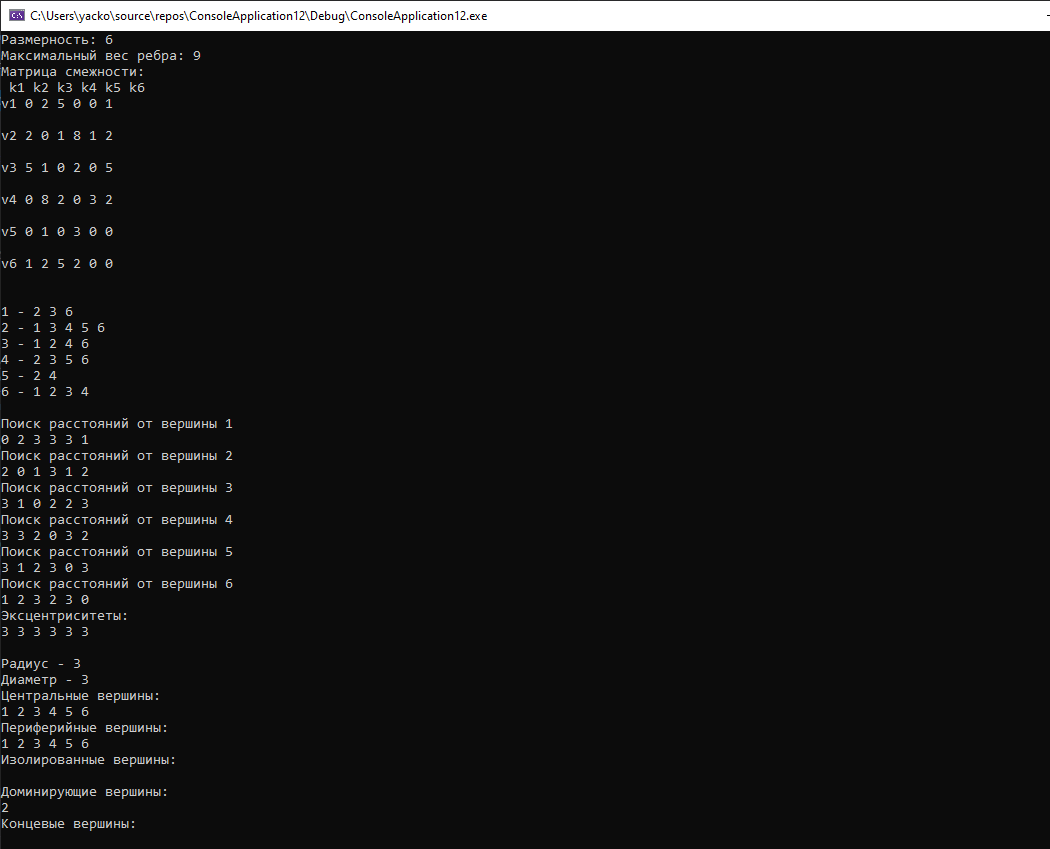
}

void main(void) {

DFS\_main();

}

**Результат работы кода**



**Вывод:** научился находить эксцентриситет вершины, определять радиус и диаметр графа, находить изолированные, концевые и доминирующие вершины.