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»

**Выполнил:**

студент группы 20ВВ3

Шадрин Данила

**Приняли:**

Юрова О.В.

Митрохин М.А.

Пенза 2021

# Название

# определение характеристик графов.

# Лабораторное задание

**Задание 1**

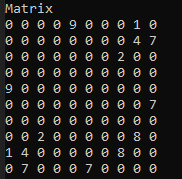
1. Сгенерируйте (используя генератор случайных чисел) матрицу смежности для неориентированного взвешенного графа *G*. Выведите матрицу на экран.
2. Определите радиус и диаметр графа *G*, используя матрицу смежности графа.
3. Определите подмножества периферийных и центральных вершин графа *G*, используя матрицу смежности.
4. Найдите изолированные, концевые и доминирующие вершины.

**Задание 2\***

1. Постройте для графа G матрицу инцидентности.
2. Определите радиус и диаметр графа *G*, используя матрицу инцидентности графа.
3. Определите подмножества периферийных и центральных вершин графа *G*, используя матрицу инцидентности.

# Результат работы программы

**Задание 1.1.** Результаты работы программы показаны на рисунке 1.



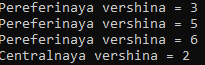
**Рисунок 1 – Результат работы программы**

**Задание 1.2.** Радиус и диаметр графа.



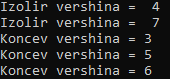
**Рисунок 2 – Результат работы программы**

**Задание 1.3.** Подмножества периферийных и центральных вершин.



**Рисунок 3 – Результат работы программы**

**Задание 1.4.** Изолированные, концевые и доминирующие вершины.



**Вывод:** яизучил алгоритмы для определения характеристик графа.

**Листинг**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

#include <conio.h>

#include <time.h>

#include <stack>

#include <vector>

#include <queue>

using namespace std;

void BFSD(int n, int\* Arr2, int i, int\*\* Arr)

{

queue<int> que;

que.push(i);

Arr2[i] = 0;

while (!que.empty())

{

i = que.front();

que.pop();

cout << i + 1 << " ";

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

{

if ((Arr[i][j] > 0) && (Arr2[j] > Arr2[i] + Arr[i][j]))

{

que.push(j);

Arr2[j] = Arr2[i] + Arr[i][j];

}

}

}

}

void main() {

system("chcp 1251"); system("cls");

int start, n;

cout << "Задание 1" << endl;

cout << "Количество вершин > "; cin >> n;

int\*\* matrix;

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

int s = 1;

srand(time(NULL));

for (int i = 0; i < n; i++) // создание матрицы

{

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

matrix[i][i] = 0;

for (int j = s; j < n; j++)

{

if (rand() % 100 > 30)

{

matrix[i][j] = 0;

}

else {

matrix[i][j] = rand() % n;

}

}

s++;

}

s = 1;

for (int i = 0; i < n; i++) // неориентированный граф

{

for (int j = s; j < n; j++)

matrix[j][i] = matrix[i][j];

s++;

}

cout << "Matrix"; // вывод матрицы на экран

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

{

cout << endl;

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

{

cout << matrix[i][j] << " ";

}

}

cout << endl;

cout << "Порядок обхода: ";

cout << endl;

//лаба 8

int r = 999999, d = -1;

int\* visited = new int[n];

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

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

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

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

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

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

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

visited[k] = 1000;

BFSD(n, visited, i, matrix);

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

{

if (extr[i] < visited[j] && visited[j] != 1000)

extr[i] = visited[j];

}

}

cout << endl;

cout << "vesa " << " ";

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

cout << extr[i] << " ";

}

cout << endl;

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

{

if (d < extr[i])

d = extr[i];

if (r > extr[i] && extr[i] != 0)

r = extr[i];

}

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

step[i] = 0;

}

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

{

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

{

if (matrix[i][j] > 0)

step[i]++;

}

}

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

isol[i] = 0;

}

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

if (step[i] == 0)

isol[i] = 1;

}

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

konz[i] = 0;

}

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

if (step[i] == 1)

konz[i] = 1;

}

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

domin[i] = 0;

}

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

if (step[i] == n - 1)

domin[i] = 1;

}

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

if (extr[i] == d)

cout << "Pereferinaya vershina = " << i + 1 << endl;

}

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

if (extr[i] == r)

cout << "Centralnaya vershina = " << i + 1 << endl;

}

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

if (isol[i] == 1)

cout << "Izolir vershina = " << i + 1 << endl;

}

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

if (konz[i] == 1)

cout << "Koncev vershina = " << i + 1 << endl;

}

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

if (domin[i] == 1)

cout << "Dominir vershina = " << i + 1 << endl;

}

cout << endl;

cout << "Diametr = " << d << endl;

cout << "Radius = " << r << endl;

cout << endl;

system("pause");

}