Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительной техники»

**ОТЧЕТ**

по лабораторной работе №6

по курсу «Логика и основы алгоритмизации в инженерных задачах.»

на тему: «Определение характеристик графов.»

Выполнили**:**

студенты группы 21ВВ4

Федоренко Вероника

Роганов Данила

Проверили:

Юрова О.В,

Акифьев И.В.

Пенза 2022

**Цель работы**

Определение характеристик графа.

**Лабораторное задание**

**Задание 1**

1. Сгенерируйте (используя генератор случайных чисел) матрицу смежности для неориентированного графа *G*. Выведите матрицу на экран.
2. Определите размер графа *G*, используя матрицу смежности графа.
3. Найдите изолированные, концевые и доминирующие вершины.

Листинг

#include <iostream>

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

bool checkIsVertertexAdjacent(int\*\* sourceMatrix, int firstVertexToDelite, int secondVertexToDelite) {

if (sourceMatrix[firstVertexToDelite][secondVertexToDelite] == 1) {

return true;

}

return false;

}

void allocateMatrix(int\*\*\* sourceMatrix, int columnCount, int rowsCount) {

\*sourceMatrix = (int\*\*)malloc(columnCount \* sizeof(int\*));

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

{

(\*sourceMatrix)[i] = (int\*)calloc(rowsCount, sizeof(int));

}

}

void fillMatrixRandomElements(int\*\* sourseMatrix, int columnCount, int rowsCount) {

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

{

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

{

if (i == j) {

sourseMatrix[i][j] = 0;

}

else

{

sourseMatrix[i][j] = rand() % 3;// & 2

sourseMatrix[j][i] = sourseMatrix[i][j];

}

//sourseMatrix[i][j] = rand() % 100 + 1; // заполняем массив случайными числами

}

}

}

void printMatrix(int\*\* sourseMatrix, int columnCount, int rowsCount) {

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

{

printf("\n");

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

{

printf("\t%d", sourseMatrix[i][j]);

}

}

}

int calculateSizeGraph(int\*\* sourseMatrix, int columnCount, int rowsCount) {

int summer = 0;

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

{

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

{

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

summer ++;

}

}

}

return summer / 2;

}

void printPowVertix(int\*\* sourseMatrix, int columnCount, int rowsCount) {

int summer = 0;

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

{

summer = 0;

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

{

if (sourseMatrix[i][j] !=0)

{

summer++;

}

}

printf("\nНомер вершины %d Степень вершины равна: %d ",i ,summer);

if (summer == 0)

{

printf(" - изолированая");

}

if (summer == 1)

{

printf(" - концевая");

}

if (summer == columnCount - 1) {

printf(" - доминирующая");

}

}

}

int\*\* createAdjacentMatrix(int\*\* sourseMatrix, int columnCount, int rowsCount) {

int\*\* resultMatrix = NULL;

int countEdges = calculateSizeGraph(sourseMatrix, columnCount, rowsCount);

allocateMatrix(&resultMatrix, columnCount, countEdges);

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

{

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

{

}

}

return resultMatrix;

}

int main()

{

setlocale(LC\_ALL, "Rus");

int\*\* firstMatrix = NULL;

int firstCount = 0;

int sizeGraph = 0;

srand(time(NULL));

printf("Введите количество первой матрицы смежности = ");

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

printf("\tM1: \n");

allocateMatrix(&firstMatrix, firstCount, firstCount);

fillMatrixRandomElements(firstMatrix, firstCount, firstCount);

printMatrix(firstMatrix, firstCount, firstCount);

printf("\n");

sizeGraph = calculateSizeGraph(firstMatrix, firstCount, firstCount);

printf("\n");

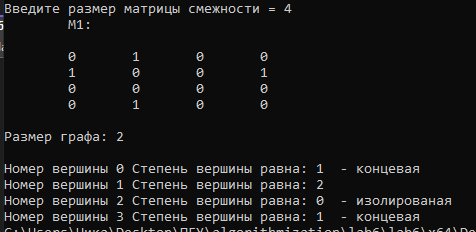
printf("Размер графа: %d", sizeGraph);

printf("\n");

printPowVertix(firstMatrix, firstCount, firstCount);

}

Результат



Вывод: в ходе выполнения лабораторной работы была разработана программа, определяющая характеристики графа.