Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №5

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Определение характеристик графов»

**Выполнили студенты**

**группы 21вв1.1:**

Вартанов А.

**Приняли:**

Юрова О.В.

Акифьев И.В.

Пенза, 2022

**Задание 1:**

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

### **Задание 2\***

1. Постройте для графа G матрицу инцидентности.
2. Определите размер графа *G*, используя матрицу смежности графа.
3. Найдите изолированные, концевые и доминирующие вершины.

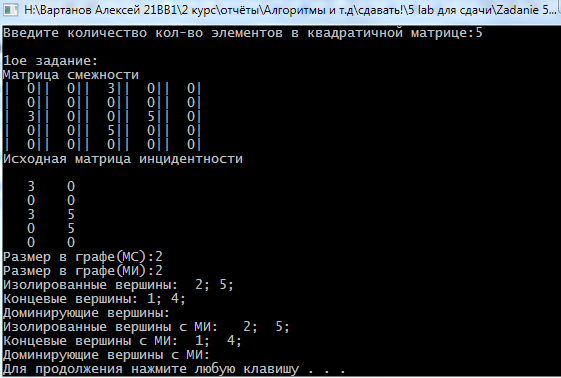
**Результат работы:**

Программа генерирует матрицу смежности и инцидентности;

Определяет размер вершин с МС и МИ;

Находит строки, в которой присутствуют изолированные, концевые и доминирующие вершины.

Программа полностью выполняет поставленную задачу.



**Вывод:**

Я научился создавать матрицы смежности и инцидентности, определять размер графа и находить разные виды вершин как: изолированные, концевые и доминирующие.

**Листинг:**

**Задание 1 и 2:**

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <locale.h>

#include <Windows.h>

#include <malloc.h>

#include <time.h>

int main()

{

setlocale(LC\_ALL,"rus");

int \*\*M;

const int razm = 100;

int n,i,j,v,k, coutRazmer = 0, m\_i[razm];

char \*inced[razm]; // Матрица инцендентности

printf("Введите количество кол-во элементов в квадратичной матрице:");

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

printf("\n1ое задание:");

//Генерация динамического массива для создания Матрицы смежности

srand((unsigned) time(NULL));

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

printf\_s("\nМатрица смежности");

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

{

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

}

//Задание 1.2 Генерация матрицы смежности:

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

{

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

if(rand()%2){

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

}

else

M[i][j]=0;

if(i==j){

M[i][j]=0;

}

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

}

}

//Вывод матрицы смежности:

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

printf\_s("\n");

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

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

}

}

//Задание 2.1 Создание матрицы инцидентности и её вывод:

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

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

if (M[i][j])

v++; // Нашли v - число вершин

}

}

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

inced[i] = (char \*) malloc(v);

memset(inced[i], 0, v);

}

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

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

if (M[i][j]){

inced[i][k] = inced[j][k] = M[i][j];

k++;}

}

}

printf("\nИсходная матрица инцидентности ");

printf("\n");

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

{

printf("\n");

m\_i[i]=inced[i][j];

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

{

if(inced[i][j]>m\_i[i])

m\_i[i]=inced[i][j];

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

}

}

//Задание 1.2 Размер графа:

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

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

if( M[i][j]>0){

coutRazmer+=1;

}

}

}

printf("\nРазмер в графе(МС):%d",coutRazmer);

//Задание 2.2 Размер графа:

for(i=0;i<1;i++){

coutRazmer=0;

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

if(inced[i][j]>= 0)

coutRazmer++;

}

}

printf("\nРазмер в графе(МИ):%d",coutRazmer);

//Задание 1.3 Виды вершин:

printf("\nИзолированные вершины: ");

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

{

int piece = 0;

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

{

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

}

if(piece==0)

printf("%2d;",i+1);

}

printf("\nКонцевые вершины:");

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

{

int piece = 0;

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

{

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

}

if(piece==1)

printf("%2d;",i+1);

}

printf("\nДоминирующие вершины: ");

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

{

int piece = 0;

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

{

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

}

if(piece==n-1)

printf("%2d;",i+1);

}

//Задание 2.3:

printf("\nИзолированные вершины c МИ: ");

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

{

int piece = 0;

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

{

if(inced[i][j]!=0) piece++;

}

if(piece==0)

printf(" %2d;",i+1);

}

printf("\nКонцевые вершины с МИ:");

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

{

int piece = 0;

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

{

if(inced[i][j]!=0) piece++;

}

if(piece==1)

printf(" %2d;",i+1);

}

printf("\nДоминирующие вершины с МИ:");

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

{

int piece = 0;

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

{

if(inced[i][j]!=0) piece++;

}

if(piece==n-1)

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

}

printf("\n");

system("Pause");

return 0;

}