МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра «Вычислительной техники»

**ОТЧЁТ**

по лабораторной работе №3

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Динамические списки»

Выполнил:

Студент группы 23ВВВ2

Монин Иван

Приняли:

доцент Юрова О.В.

доцент Митрохин М.А.

Пенза 2024

**Цель работы** – научиться создавать различные сложные динамические структуры данных.

**Лабораторное задание:**

**Задание 1**

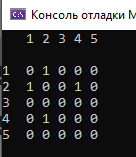
1. Сгенерируйте (используя генератор случайных чисел) матрицу смежности для неориентированного графа *G*. Выведите матрицу на экран.
2. Определите размер графа *G*, используя матрицу смежности графа.
3. Найдите изолированные, концевые и доминирующие вершины.

### Задание 2\*

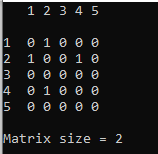
1. Постройте для графа G матрицу инцидентности.
2. Определите размер графа *G*, используя матрицу инцидентности графа.
3. Найдите изолированные, концевые и доминирующие вершины.

**Задание 1**

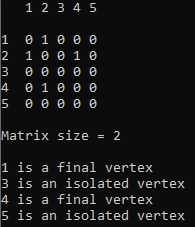
1. Сначала выводится сама матрица смежности



1. Затем выводится размер матрицы (кол-во рёбер графа). Так как это матрица смежности, то мы считаем все «1», которые выше главной диагонали

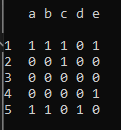


1. После размера матрицы выводится вся информация о вершинах (конечные, изолированные и доминирующие вершины)

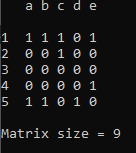


**Задание 2**

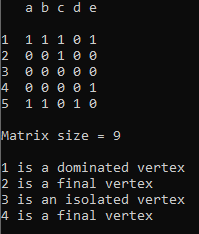
1. Сначала выводится сама матрица инцидентности



1. Затем выводится размер матрицы (кол-во рёбер графа). Так как это матрица инцидентности, то мы считаем все «1»



1. После размера матрицы выводится вся информация о вершинах (конечные, изолированные и доминирующие вершины)



**Листинг:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

void main() {

srand(time(NULL));

int matrix[5][5];

int matrixSize = 0;

int degree = 0;

for (int i = 0; i < 5; i++) {

for (int j = i; j < 5; j++) {

matrix[i][j] = rand() % 2;

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

if (i == j)matrix[i][i] = 0;

if (matrix[i][j] == 1) {

matrixSize++;

}

}

}

printf(" 1 2 3 4 5\n\n");

for (int i = 0; i < 5; i++) {

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

for (int j = 0; j < 5; j++) {

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

}

printf("\n");

}

printf("\nMatrix size = %d\n\n", matrixSize);

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 5; j++) {

if (matrix[i][j] == 1) {

degree++;

}

}

if (degree == 0) printf("%d is an isolated vertex\n", i + 1);

if (degree == 1) printf("%d is a final vertex\n", i + 1);

if(5-degree==1)printf("%d is a dominated vertex\n", i+1);

degree = 0;

}

matrixSize = 0;

////////

int maxOne = 2;

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 5; j++) {

matrix[j][i] = rand() % 2;

if (matrix[j][i] == 1) {

if (maxOne > 0) maxOne--;

else matrix[j][i] = 0;

}

if (j == 4 && maxOne == 1) matrix[j][i] = 1;

if (matrix[j][i] == 1) {

matrixSize++;

}

}

maxOne = 2;

}

printf("\n\n a b c d e\n\n");

for (int i = 0; i < 5; i++) {

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

for (int j = 0; j < 5; j++) {

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

}

printf("\n");

}

printf("\nMatrix size = %d\n\n", matrixSize);

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 5; j++) {

if (matrix[i][j] == 1) {

degree++;

}

}

if (degree == 0) printf("%d is an isolated vertex\n", i + 1);

if (degree == 1) printf("%d is a final vertex\n", i + 1);

if (5 - degree == 1)printf("%d is a dominated vertex\n", i + 1);

degree = 0;

}

}

**Вывод**

В ходе работы выполнения лабораторной работы было реализовано создание матрицы инцидентности и смежности. Также были рассчитаны их размеры и была найдена информация о вершинах графа.