Пензенский государственный университет

Кафедра «Вычислительная техника»

**Отчёт**

По лабораторной работе №4

по дисциплине: «Логика и основы оптимизации в инженерных задачах»

## на тему «Обход графа в глубину»

Выполнил студент группы 19ВВ4:

Исаев Д.

Приняли:

Митрохин М. А.

Юрова О.В.

Пенза 2020

**Цель работы:** освоить алгоритм обхода графа в глубину на матрице.

**Общие сведения:**

Обход графа – одна из наиболее распространенных операций с

графами. Задачей обхода является прохождение всех вершин в графе.

Обходы применяются для поиска информации, хранящейся в узлах графа,

нахождения связей между вершинами или группами вершин и т.д.

Одним из способов обхода графов является поиск в глубину. Идея

такого обхода состоит в том, чтобы начав обход из какой-либо вершины

всегда переходить по первой встречающейся в процессе обхода связи в

следующую вершину, пока существует такая возможность. Как только в

процессе обхода исчерпаются возможности прохода, необходимо вернуться

на один шаг назад и найти следующий вариант продвижения. Таким образом,

итерационно выполняя описанные операции, будут пройдены все доступные

для прохождения вершины. Чтобы не заходить повторно в уже пройденные

вершины, необходимо их пометить как пройденные.

Таким образом, можно предложить следующую рекурсивную

реализацию алгоритма обхода в глубину.

Реализация состоит из подготовительной части, в которой все вершины

помечаются как не помеченные и осуществляется запуск процедуры

обхода для вершин графа . И непосредственно процедуры обхода,

которая помечает текущую (т.е. ту, в которой на текущей итерации находится

алгоритм) вершину как посещенную . Затем выводит номер текущей

вершины на экран и в цикле просматривает v-ю строку матрицы

смежности графа G(v,i). Как только алгоритм встречает смежную с v не

посещенную вершину , то для этой вершины вызывается процедура

обхода.

**Практическая часть**

**Задание 1.1, 1.2**: Сгенерировали (используя генератор случайных чисел) матрицу смежности для неориентированного графа G. Для сгенерированного графа осуществили процедуру обхода в глубину.

**Листинг.**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <locale.h>

#include <time.h>

int\*\* a;

int\* vis;

void DFS(int s, int rows){

vis[s] = 1; printf("%d", s);

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

if(a[s][i] == 1 && vis[i] == 0){

DFS(i, rows);

}

}

}

int main(){

setlocale(LC\_ALL, "Rus");

srand(time(NULL));

int rows;

int nomer;

int n = 0;

int k = 0;

system("cls");

printf("Укажите размер массива: ");

scanf("%d", &rows);

printf("Введите номер для прохождения: ");

scanf("%d", &nomer);

a = (int\*\*)malloc(rows \* sizeof(int\*));

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

a[i] = (int\*)malloc(rows \* sizeof(int));

}

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

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

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

if(i == j){

a[i][j] = 0;

}

a[j][i] = a[i][j];

}

}

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

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

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

}

printf("\n");

}

printf("\n");

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

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

vis[i] = 0;

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

}

printf("\n");

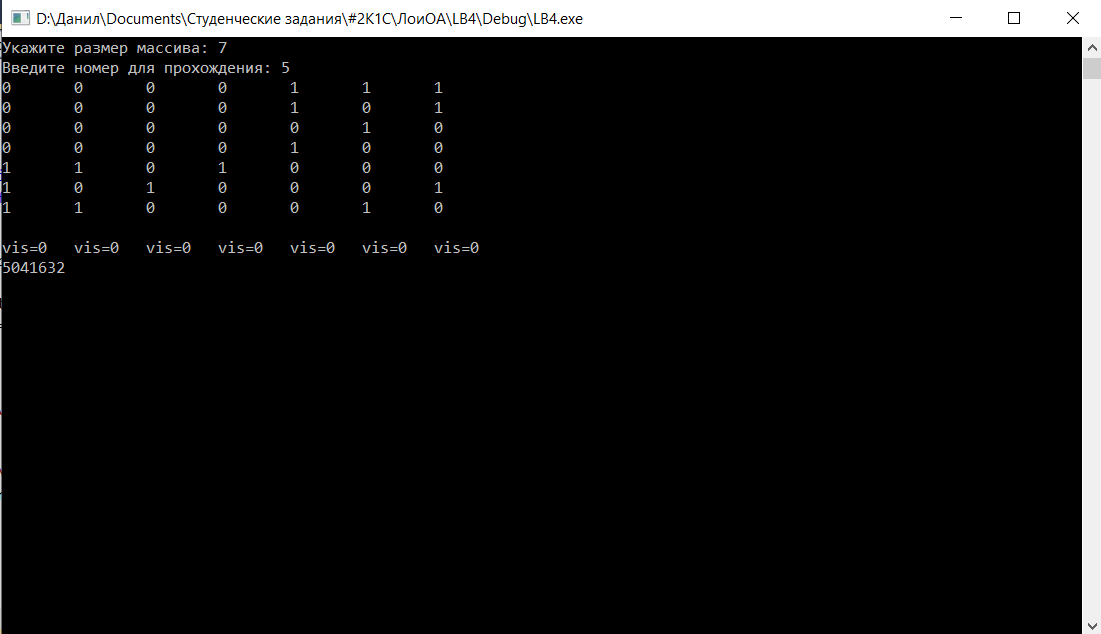
DFS(nomer,rows);

getch();

return 0;

}

## Результат работы.



**Вывод:** освоили алгоритм обхода графа в глубину на матрицах. Научились реализовывать алгоритм обхода графа с использованием рекурсии.