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:

Земсков М.А.

Приняли:

Митрохин М. А.

Юрова О.В.

Пенза 2020

**Цель работы:** освоить алгоритм обхода графа в глубину на матрицах. Научиться реализовывать алгоритм обхода графа с использованием рекурсии.

**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**

1. Сгенерировать (используя генератор случайных чисел) матрицу

смежности для неориентированного графа G. Вывести сгенерированные

матрицы на экран.

2. Для сгенерированного графа осуществить процедуру обхода в

глубину, реализованную в соответствии с приведенным выше описанием.

**Листинг.**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <stdio.h>

#include <conio.h>

#include <malloc.h>

#include <stdlib.h>

#include <time.h>

int\*\* a;

int\* vis;

int ver;

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()

{

int rows;

int n = 0;

int k = 0;

system("cls");

printf("rows:");

scanf("%d", &rows);

srand(time(NULL));

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 = 0; 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 ", a[i][j]);

}

printf("\n");

}

printf("\n");

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

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

vis[i] = 0;

}

printf("\n");

printf("Vershina:");

scanf("%d", &ver);

printf("\n");

DFS(ver,rows);

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

free(a[i]);

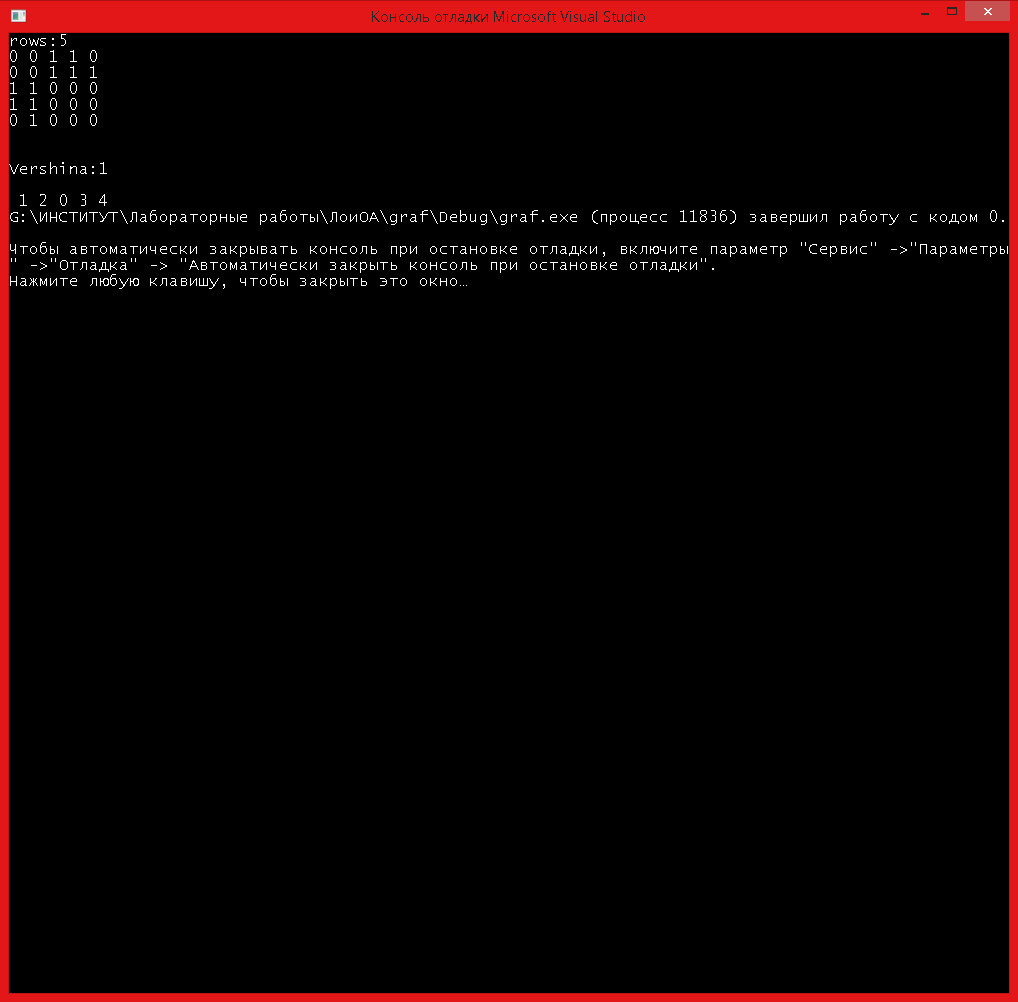
free(a);

getchar();

return 0;

}

## Результат работы.



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