Министерство науки и высшего образования РоссийскойФедерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №7

по теме «Обход графа в глубину»

Выполнили:

Студенты группы 21ВВ2

Изосин А.А.  
Волдырев В.С.

Принял:

Митрохин М.А.

Юрова О.В.

Пенза, 2022

**Цель работы:**

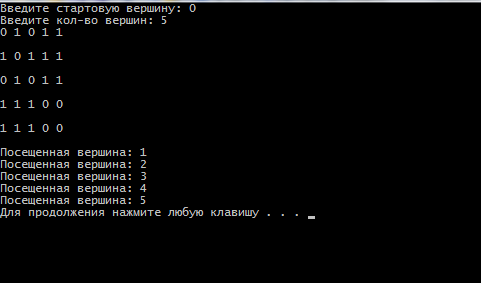
Изучить основные алгоритмы обхода графа и научиться решать задачи обхода графа на основе поиска в глубину.

**Лабораторные работы:**

**Задание 1**

1. Сгенерируйте (используя генератор случайных чисел) матрицу смежности для неориентированного графа *G*. Выведите матрицу на экран.
2. Для сгенерированного графа осуществите процедуру обхода в глубину, реализованную в соответствии с приведенным выше описанием.

**Результат работы программы:**



**Листинг:**

**DFS**

#include "stdafx.h"

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <Windows.h>

#include <iostream>

using namespace std;

void task1(int\*\* matrix, int edgenum)

{

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

{

matrix[i] = (int\*)malloc(edgenum \* sizeof(int));

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

{

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

}

}

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

{

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

{

if (i == j)

{

matrix[i][j] = 0;

}

else

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

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

}

printf("\n");

printf("\n");

}

}

void DFS(int\*\* matrix, bool\* visited, int start, int edgenum)

{

visited[start] = true;

printf("Посещенная вершина: %d ", start + 1);

printf("\n");

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

{

if ((matrix[start][i] != 0) && (!visited[i]))

{

DFS(matrix, visited, i, edgenum);

}

}

}

int main()

{

setlocale(0, "");

srand(time(NULL));

int start;

printf("Введите стартовую вершину: ");

scanf("%d", &start);

int edgenum;

printf("Введите кол-во вершин: ");

scanf("%d", &edgenum);

int\*\* matrix = (int\*\*)malloc(edgenum \* sizeof(int\*\*));

bool\* visited = (bool\*)malloc(edgenum \* sizeof(bool\*));

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

{

visited[i] = false;

}

task1(matrix, edgenum);

DFS(matrix, visited, start, edgenum);

system("pause");

return 0;

}

**Вывод:** Изучили основные алгоритмы обхода графа и научились решать задачи обхода графа на основе поиска в глубину.