

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра «Вычислительная техника»

ОТЧЕТ

по лабораторной работе №7

по дисциплине: "Логика и основы алгоритмизации в инженерных задачах"

на тему: "Обход графа в глубину"

Выполнили студенты группы
24ВВВЗ:

Пяткин Р. С.

Гусаров Е. Е.

Принял:

к.т.н., доцент, Юрова О. В.

к.т.н., Деев М. В.

Пенза 2025

Цель

Изучение обхода графа в глубину.

Лабораторное задание

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.

Результаты программ

```
Введите кол-во вершин в матрице: 8
  0  1  2  3  4  5  6  7
0| 0  1  1  1  0  0  1  1
1| 1  0  1  0  0  1  0  0
2| 1  1  0  0  0  1  0  1
3| 1  0  0  0  1  1  1  0
4| 0  0  0  1  0  0  0  0
5| 0  1  1  1  0  0  0  0
6| 1  0  0  1  0  0  0  0
7| 1  0  1  0  0  0  0  0

Вершина с которой начинать обход: 1
Результат: 10253467
```

Рис. 2 — lab_5_2

Вывод:

В ходе выполнения лабораторной работы были разработаны программа для выполнения заданий Лабораторной работы №7. В процессе выполнения работы был изучен способ обхода графа в глубину.

Листинг

Файл lab_7.cpp

```
#include <iostream>
#include <ctime>
#include <random>

using namespace std;

int printMatrix(int** m, int n){
    cout << " ";
    for(int i = 0; i < n; i++){
        cout << i << " ";
    }
    cout << endl;
    for(int i = 0; i < n; i++){
        cout << i << "| ";
        for(int j = 0; j < n; j++){
            cout << m[i][j] << " ";
        }
        cout << endl;
    }
    cout << endl;
    return 0;
}

int** createMatrix(int n){
    int** m = new int*[n];
    for (int i = 0; i < n; i++) {
        m[i] = new int[n];
    }

    for(int i = 0; i < n; i++){
        for(int j = i; j < n; j++){
            m[i][j] = m[j][i] = (i == j ? 0 : rand() % 2);
        }
    }
    return m;
}

void deleteM(int** m, int n, int* v) {
    for (int i = 0; i < n; ++i) {
        delete[] m[i];
    }
    delete[] m;
    delete[] v;
}

void DFS(int** G, int numG, int* visited, int s){
    visited[s] = 1;
    cout<<s;
    for(int i = 0; i < numG; i++){
```

```
if(G[s][i] == 1 && visited[i] == 0){  
    DFS(G, numG, visited, i);  
}  
}  
}
```

```
int main(){  
    srand(time(0));  
    int numG, current = 0;  
    cout << "Введите кол-во вершин в матрице: ";  
    cin >> numG;  
    int* visited = new int[numG];
```

```
    for(int i = 0; i < numG; i++){  
        visited[i] = 0;  
    }
```

```
    int** G = createMatrix(numG);  
    printMatrix(G, numG);
```

```
    cout << "Вершина с которой начинать обход: ";  
    cin >> current;  
    cout << "Результат: ";  
    DFS(G, numG, visited, current);  
    cout << endl;  
    deleteM(G, numG, visited);  
    return 0;  
}
```