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 21ВВ4

Гребенев А.С.

Никишин Д.Д.

Принял:

Акифьев И. В.  
Юрова О. В.

Пенза, 2022

**Цель работы:** разработать программный код для реализации обхода графа в глубину.

**Лабораторная работа:**

**Задание 1.** Сгенерируйте (используя генератор случайных чисел) матрицу смежности для неориентированного графа *G*. Выведите матрицу на экран.

Для сгенерированного графа осуществите процедуру обхода в глубину, реализованную в соответствии с приведенным выше описанием.

#include <iostream>

#include <locale>

#include <stdio.h>

#define \_CRT\_SECURE\_NO\_WARNINGS

using namespace std;

void DFS(int v, int n, int\* M[], bool vis[])

{

vis[v] = 1;

cout << v << "\t";

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

{

if (M[v][i] == 1 && vis[i] == 0) {

DFS(i, n, M, vis);

}

}

}

int main()

{

setlocale(LC\_ALL, "rus");

int n;

cout << "Введите размерность квадратной матрицы смежности: ";

cin >> n;

int\*\* matrix = new int\* [n];

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

{

matrix[i] = new int[n];

}

srand(time(NULL));

cout << "\nМатрица смежности:" << endl;

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

{

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

{

matrix[i][j] = rand() % 100;

if (i == j) {

matrix[i][j] = 0;

}

if (matrix[i][j] >= 0 && matrix[i][j] <= 20)

{

matrix[i][j] = 0;

}

else if (matrix[i][j] >= 21 && matrix[i][j] < 100)

{

matrix[i][j] = 1;

}

}

}

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

{

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

{

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

cout << matrix[i][j] << " ";

}

cout << endl;

}

int v;

cout << "\nВведите вершину от 0 до "<< n-1 << ": ";

cin >> v;

cout << endl;

bool\* vis = new bool[v];

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

{

vis[i] = { 0 };

}

DFS(v, n, matrix, vis);

//очищение памяти:

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

delete[] matrix[i];

}

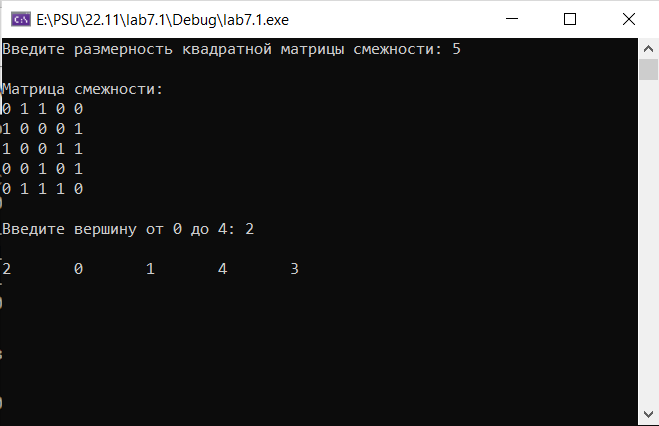
delete[] matrix;

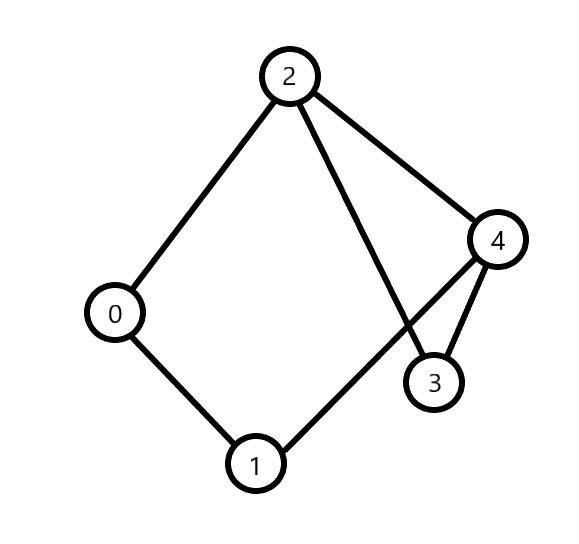
delete[] vis;

return 0;

}

Результат работы программы:





**Вывод:** мы реализовали алгоритм поиска обхода графа в глубину