Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №4

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Обход графа в глубину.»

**Выполнили студенты группы 21вв1:**

Кирьянов В.Е.

Аляев А.О.

**Приняли**

Юрова О.В.

Акифьев И.В.

Пенза 2022

### **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.

**3.**\* Реализуйте процедуру обхода в глубину для графа, представленного списками смежности.

### **Задание 2\***

1. Для матричной формы представления графов выполните преобразование рекурсивной реализации обхода графа к не рекурсивной.

**Листинг программы:**

**Задание 1.1;1.2;2.1:**

#include "stdafx.h"

#include <iostream>

#include <stack>

#include <string>

#include <limits>

#include "time.h"

using namespace std;

//const int n=5;

int i, j,size;

bool \*visited=new bool[200];

//матрица смежности графа

int \*\*graph;

int\*\* create\_mass(int size){

//gen\_create = true;

int \*\*graph = (int\*\*)malloc(sizeof(int\*)\*size);

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

graph[i] = (int\*)malloc(sizeof(int)\*size);

}

return graph;

}

int\*\* initializate(int\*\* graph, int size){

//gen\_init = true;

srand(time(NULL));

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

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

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

}

}

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

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

if(i == j){

graph[i][j] = 0;

}

if(graph[i][j] = graph[j][i]){

graph[i][j] = graph[j][i];

}

}

}

return graph;

}

//поиск в глубину

void DFS(int st)

{

int r;

cout<<st+1<<" ";

visited[st]=true;

for (r=0; r<=size; r++)

if ((graph[st][r]!=0) && (!visited[r]))

DFS(r);

}

void main()

{

setlocale(LC\_ALL, "Rus");

int start = 1;

cout << "Введите количество вершин (больше 2): ";

cin >> size;

graph = create\_mass(size);

initializate(graph,size);

cout<<"Матрица смежности графа: "<<endl;

for (i=0; i<size; i++)

{

visited[i]=false;

for (j=0; j<size; j++)

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

cout<<endl;

}

bool \*vis=new bool[size];

cout<<"Порядок обхода рекурсивной функции: ";

DFS(start-1);

delete []visited;

cout<<"\nПорядок обхода нерекурсивной функции: ";

stack <int> Stack;

int nodes[200];

for (int i = 0; i < size; i++) nodes[i] = 0;

Stack.push(0);

while (!Stack.empty())

{ // пока стек не пуст

int node = Stack.top();

Stack.pop();

if (nodes[node] == 2) continue;

nodes[node] = 2;

for (int j = size - 1; j >= 0; j--)

{

if (graph[node][j] == 1 && nodes[j] != 2)

{

Stack.push(j);

nodes[j] = 1;

}

}

cout << node + 1 <<" ";

}

system("pause>>void");

}

**Ход работы:**

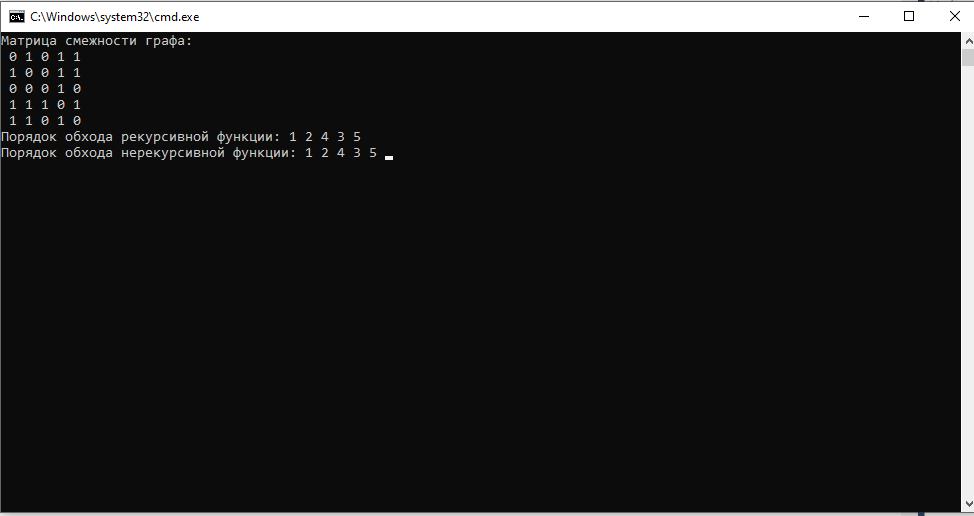
**Задание 1:**

Рисунок 1 – результат работы программы.

**Проверка:**

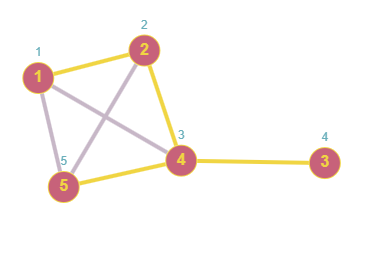
****

Рисунок 2 – результат проверки.

**Вывод:** в ходе выполнения лабораторной работы были разработаны программы, выполняющие обход графа в глубину.

Получили опыт в создании проектов в среде Microsoft Visual Studio, научились писать и отлаживать программы с применением поиска в глубину на языке Си.