Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЁТ**

по лабораторной работе №4

по курсу «Логика и основы алгоритмизации в инженерных задачах»

на тему «Обход графа в глубину»

Выполнили:

студенты группы 20ВВ2

Киреев Б.П.

Верховский М.В.

Лукин В.Д.

Приняли:

д.т.н. Митрохин М.А.

к.т.н. доцент Юрова О.В.

Пенза 2021

Название: Обход графа в глубину.

Цель работы: Научиться осуществлять обход 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.

3. Реализуйте процедуру обхода в глубину для графа, представленного

списками смежности.

**Задание 1:**

Листинг:

Файл Lab4.cpp

#include <iostream>

#include <ctime>

#include <malloc.h>

#include <conio.h>

using namespace std;

int\* vis = NULL, n, \*\* p = NULL;

void DFS(int num) {

cout << num << " ";

vis[num] = 1;

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

if (p[num][i] == 0 && vis[i] == 0) {

DFS(i);

}

}

}

int main() {

setlocale(LC\_ALL, "Rus");

srand(time(0)); //по-настоящему случайная генерация массива

cout << "Введите размер матрицы: ";

cin >> n;

cout << "Массив: \n";

vis = (int\*)calloc(n, 3);

p = (int\*\*)calloc(n, 3);

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

p[i] = (int\*)calloc(n, 3);

}

cout.precision(3 \* n);

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

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

p[i][m] = 0 + rand() % 2;

p[i][i] = 0;

cout << p[i][m] << " ";

}

cout << endl;

}

cout << "Вершины: " << endl;

DFS(0);

return 0;

}

Пояснительный текст к программе:

Данная программа осуществляет обход графа в глубину. Работа алгоритма продолжается до тех пор, пока существуют непосещённые вершины.

Результаты работы программы:

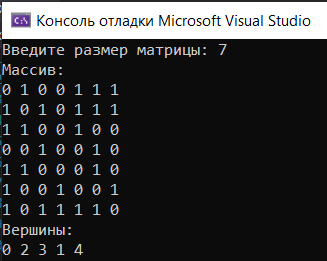


Рисунок №1 – Работа программы из задания №1, пунктов 1 и 2.

**Задание 2:**

Листинг:

Файл Lab4\_3.cpp

#include <iostream>

#include <ctime>

#include <malloc.h>

#include <conio.h>

using namespace std;

int\* vis = NULL, n, \*\* p = NULL, \*\* x = NULL;

int pogr = 1;

void DFS(int num) {

cout << num + 1 << " "; // вывод вершин

vis[num] = 1;//вершина пройдена

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

if (p[num][i] == 0 && vis[i] == 0) { // проверка на посещение ранее

DFS(i);

}

}

}

int main() {

setlocale(LC\_ALL, "Rus");

srand(time(0)); //по-настоящему случайная генерация массива

cout << "Пункты 1, 2:\n";

cout << "Введите размер матрицы: ";

cin >> n;

cout << "Массив: \n";

vis = (int\*)calloc(n, 3);

p = (int\*\*)calloc(n, 3);

x = (int\*\*)calloc(n, 3);

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

p[i] = (int\*)calloc(n, 3);

x[i] = (int\*)calloc(n, 3);

x[i][0] = i + 1;

}

cout.precision(3 \* n);

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

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

p[i][m] = 0 + rand() % 2;

if (p[i][m] == 1 && i!=m) { x[i][pogr] = m + 1; pogr++; }

else { x[i][pogr] = 0; pogr++; }

p[i][i] = 0;

cout << p[i][m] << " ";

}

pogr = 1;

cout << endl;

}

cout << "Вершины: " << endl;

DFS(0);

cout << "\n\nПункт 3:\n";

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

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

if (x[i][m]!=0){cout << x[i][m] << " ";}

}

cout << endl;

}

return 0;

}

Пояснительный текст к программе:

Данная программа осуществляет обхода в глубину для графа, представленного

списками смежности.

Результаты работы программы:

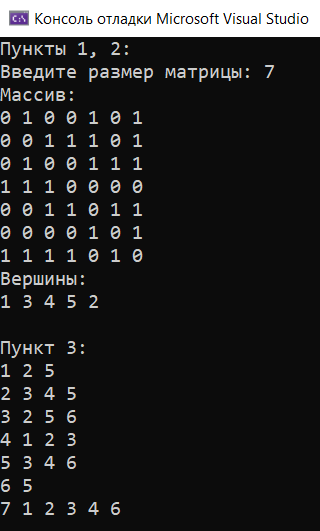


Рисунок №2 – Работа программы из задания №1, пункта 3.

Вывод: При выполнении данной лабораторной работы мы смогли научиться осуществлять обход графа в глубину.