Пензенский государственный университет

Кафедра "Вычислительной техники"

**Отчёт**

По лабораторной №4

по дисциплине "ЛиОАвИТ"

на тему "Обход графа в глубину"

**Выполнили студенты группы 19ВВ1:**

Буданов Д.

**Приняли:**

Митрохин М. А.

Юрова О. В.

Пенза 2020

# Цель работы: выполнить обход графа в глубину.

**Лабораторное задание:**

Задание 1

1. Сгенерируйте (используя генератор случайных чисел) две матрицы

смежности для неориентированного графа G. Выведите сгенерированные

матрицы на экран.

2. Для сгенерированного графа осуществите процедуру обхода в

глубину, реализованную в соответствии с приведенным выше описанием.

**Теоретическая часть:**

Обход графа – одна из наиболее распространенных операций с

графами. Задачей обхода является прохождение всех вершин в графе.

Обходы применяются для поиска информации, хранящейся в узлах графа,

нахождения связей между вершинами или группами вершин и т.д.

Одним из способов обхода графов является поиск в глубину. Идея

такого обхода состоит в том, чтобы начав обход из какой-либо вершины,

всегда переходить по первой встречающейся в процессе обхода связи в

следующую вершину, пока существует такая возможность. Как только в

процессе обхода исчерпаются возможности прохода, необходимо вернуться

на один шаг назад и найти следующий вариант продвижения. Таким образом,

итерационно выполняя описанные операции, будут пройдены все доступные

для прохождения вершины. Чтобы не заходить повторно в уже пройденные

вершины, необходимо их пометить как пройденные.

**Практическая часть:**

**Листинг:**

#include "stdafx.h"

#include <iostream>

#include <stdio.h>

#include <locale>

#include <malloc.h>

#include <stdlib.h>

#include <time.h>

#include <conio.h>

int \*\*mass, \*visit;

void DFS(int p, int n) {

int s;

printf("%d ", p);

visit[p] = 1;

for (s = 0; s < n;s++) {

if (visit[s] == 0 && mass[p][s] == 1) {

DFS(s, n);

}

}

}

int main() {

setlocale(0, "rus");

int p, k = 0, n;

printf("Введите размер матрицы:");

scanf\_s("%d", &n);

mass = (int \*\*)malloc(n\*sizeof(int));

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

mass[i] = (int \*)malloc(n\*sizeof(int));

}

visit = (int \*)malloc(n\*sizeof(int));

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

visit[i] = 0;

}

printf("Пути в графе:\n");

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

for (int j = 0;j < i + 1;j++) {

k = rand() % 101;

if (k >= 20) {

if (i != j) {

mass[i][j] = 0;

mass[j][i] = 0;

}

else {

mass[i][j] = 0;

}

}

else if (k < 20) {

if (i != j) {

mass[i][j] = 1;

mass[j][i] = 1;

}

else {

mass[i][j] = 0;

}

}

}

}

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

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

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

}

printf("\n");

}

printf("\n");

for (p = 0; p < n;)

{

printf("\n");

DFS(p, n);

p++;

}

/\*for (int i = 0;i<1;i++) {

printf("Введите вершину с которой начнется обход:");

scanf\_s("%d", &p);

if (p <= n) {

printf("\nВы проходите");

}

else {

printf("Выберете новую координату\n");

i--;

}

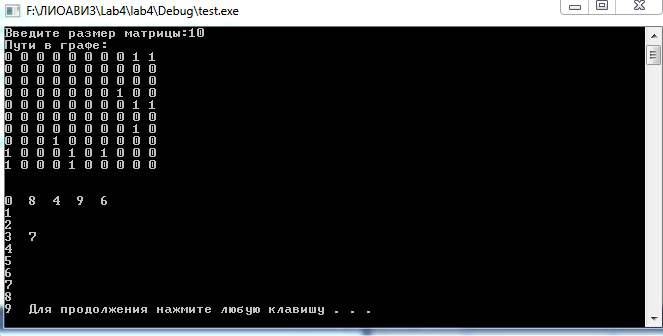
}\*/

getchar();

system("pause");

}

**Результат работы программы:**



**Вывод:** Научились выполнять обход графа в глубину.