|  |
| --- |
| Министерство образования Российской Федерации  Пензенский государственный университет  Кафедра «Вычислительная техника» |
| Отчет  по лабораторной работе №4  по курсу «Л и ОА в ИЗ»  на тему «Обход графа в глубину» |
|  |
|  |
| Выполнил:  студент группы 20ВВ4  Кривцов Н.А.  Слабинский Н.А.  Приняли:  Юрова О.В.  Акифьев И.В. |
| Пенза 2021 |

# Цель работы: выполнить обход графа в глубину.

**Лабораторное задание:**

Задание 1

1. Сгенерируйте (используя генератор случайных чисел) две матрицу

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.

**Теоретия:**

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.

**Листинг:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "locale"

#include "stdio.h"

#include "conio.h"

#include <locale.h>

#include <stdlib.h>

#include <malloc.h>

#include <iostream>

int NUM[6] = { 0,0,0,0,0,0 };

void DFS(int v, int\*\* matrix, int size) {

NUM[v] = 1;

printf("%d", v);

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

if (matrix[v][i] == 1 && NUM[i] == 0)

{

DFS(i, matrix, size);

}

}

}

int main()

{

srand(time(NULL));

setlocale(LC\_ALL, "RUS");

int v, n;

int k = 0;

printf("Введите размер матрицы: ");

scanf("%d", &n);

int\*\* M;

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

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

{

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

}

printf("Матрицы:");

printf("\n");

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

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

if (j > k) {

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

printf("%3d", M[i][j]);

}

else if (j < k) {

M[i][j] = M[j][i];

printf("%3d", M[i][j]);

}

else {

M[i][j] = 0;

printf("%3d", M[i][j]);

}

}

k++;

printf("\n");

}

printf("\n\n");

printf("Введите вершину, с которой будет начинаться обход: ");

scanf("%d", &v);

printf("\n");

printf("Результат: ");

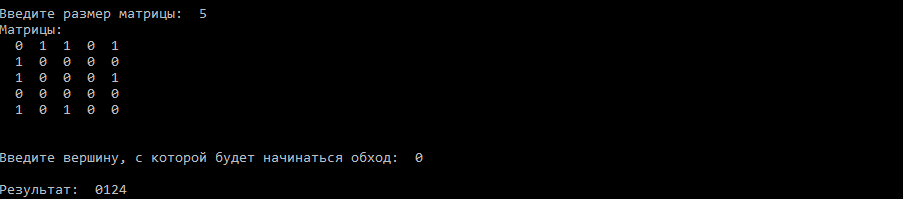
while (NUM[v] != 1)

{

DFS(v, M, n);

}

}

**Вывод:** Разобрали метод обхода графа в глубину и написали код который выполняет это автоматически.