Министерство науки и высшего образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №7

по курсу “Логика и основа алгоритмизации в ИЗ”

на тему “Обход графа в глубину ”

Выполнили студенты группы 21ВВ3:

Савосин Владислав

Юсеев Руслан

Димитренко Никита

Приняли:

Митрохин М.А. , Юрова О.В.

Пенза 2022

**Название:**

Обход графа в глубину.

**Цель работы:** Сделать рекурсивный алгоритм обход графа в глубину.

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.

Таким образом, можно предложить следующую рекурсивную реализацию алгоритма обхода в глубину.

**Лабораторное задание:**

1. Задание 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.

**Описание программы:**

В данном задании мы делали обход графа в ширину. Мы создали рекурсивную функцию DFS, главной задачей которой, является отслеживание иллюстрация хранение посещенных вершин графа, а так же вызов этой функции. В функции main мы задаём массив заполненный 0 иллюстрация 1.

#include <ctime>

#include <stack>

#include <iostream>

#include "windows.h"

using namespace std;

//граф | посещ.вершины |кол-во вершин |данная вершина

void DFS(int\*\* arr, int\* visited, int size, int top) {

if (visited[top] == 1) return; // если вершина уже посещена

visited[top] = 1; // если не посещена то отмечаем как посещенную

cout << top << ' ';

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

if (arr[top][i] == 1 && visited[i] != 1) { // если нашли ребро с вершиной, которая еще не посещена

DFS(arr, visited, size, i); // вызываем функцию для это вершины

}

}

}

// функция вызывает DFS для каждой вершины графа

void vivod(int\*\* arr, int size, int i = 0) {

if (i == size) return; // конец функции

cout << endl << "Обход с вершины " << i << endl << endl;

int\* visited = new int[size];

for (int i = 0; i != size; i++) visited[i] = 0; // хранит посещенные вершины

DFS(arr, visited, size, i);

vivod(arr, size, ++i); // рекурсивный вызов для след. вершины

}

int main()

{

setlocale(LC\_ALL, "RUS");

srand(time(NULL));

int\*\* arr, \* visited; //граф

int size; //размер

cout << "Введите кол-во вершин: ";

cin >> size;

arr = new int\* [size];

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

arr[i] = new int[size]; // создаем массивы для нашей матрицы

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

arr[i][j] = 0; // заполняем матрицу 0

}

}

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

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

int r = rand() % 100;

if (r > 60) {

arr[i][j] = 1;

arr[j][i] = 1;

}

}

}

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

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

cout << arr[i][j] << ' ';

}

cout << endl;

}

int top;

visited = new int[size];

cout << "Введите стартовую вершину: ";

cin >> top;

cout << endl;

DFS(arr, visited, size, top);

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

visited[i] = 0;

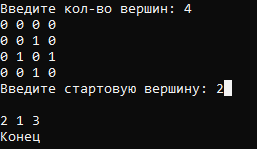
}

cout << "\nКонец" << endl;

return 0;

}

**Вывод данной программы**:

  
  
**Вывод:** Мы создали матрицу смежности для неориентированного графа иллюстрация и провели рекурсивный обход в глубину.