|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7**

**«ГРАФЫ»**

Студент Ковалец Кирилл

Группа ИУ7 – 33Б

*2020 г.*

**Описание условия задачи**

Обработать графовую структуру в соответствии с указанным вариантом задания. Обосновать выбор необходимого алгоритма и выбор структуры для представления графов. Ввод данных – на усмотрение программиста. Результат выдать в графической форме.

**Описание технического задания**

Найти все вершины заданного opграфа, недостижимые из заданной его вершины.

**Входные данные:**

Номер команды, отвечающий за определённое действие.

**Команды:**

1. Загрузить данные из файла (5 элементов в графе);
2. Загрузить данные из файла (10 элементов в графе);
3. Загрузить данные из файла (15 элементов в графе);
4. Ввести данные вручную;
5. Выйти из программы.

**Выходные данные:**

1. Результат выполнения определённой команды;
2. Печать графа с цветными вершинами (зелёный – вершина достижима из заданной вершины, красный - недостижима).

**Обращение к программе:**

Запускается через терминал командой make run

**Сообщения при аварийных ситуациях:**

1. Не удалось прочитать номер команды;
2. Номер команды должен быть >= 0 и <= 4;
3. Не удалось открыть файл;
4. Не удалось закрыть файл;
5. Не удалось выделить память;
6. Не удалось прочитать кол-во вершин графа;
7. Кол-во вершин графа должно быть больше 0;
8. Не удалось прочитать вершину графа;
9. Такой вершины в графе нет.

**Описание структуры данных**

**graph\_h** - структура, содержащая информацию об орграфе.

typedef struct graph

{

int \*\*matrix;

int size;

} graph\_h;

**Поля структуры:**

1. int \*\*matrix – матрица смежности;
2. int size - кол-во вершин в графе;

**Описание алгоритма**

1. Выводится меню программы (каждой команде присвоен номер);
2. Пользователь вводит номер команды, который отвечает за определённое действие;
3. Вывод графа на экран относительно заданной вершины осуществляется до того момента, пока не будет введён 0, являющийся признаком выхода из программы (поиск вершин в графе осуществляется обходом в глубину).

**Набор тестов**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Название теста** | **Входные данные** | **Результат** |
| 1 | Номер команды - число | k | Не удалось прочитать номер команды |
| 2 | Номер команды >= 0 и <=4 | 8 | Номер команды должен быть >= 0 и <= 4 |
| 3 | Открытие несуществующего файла | Iu7.txt | Не удалось открыть файл |
| 4 | Кол-во вершин в графе – число | k | Не удалось прочитать кол-во вершин графа |
| 5 | Кол-во вершин в графе > 0 | 0 | Кол-во вершин графа должно быть больше 0 |
| 6 | Вершина графа - число | k | Не удалось прочитать вершину графа |
| 7 | Заданная вершина должна существовать в графе | (граф состоит из 5 вершин)  10 | Такой вершины в графе нет |
| 8 | Загрузка данных из файлов | Команды 1-3 | Данные успешно загружены |
| 9 | Вывод графа на экран | Команда 4  (кол-во вершин в графе - 15)  1 10  7 3  4 4  5 9  2  6  3 8  8 10  7 7  15 7  4 7  13 2  9 12  12 8  6 9  12 5  0  (относительная вершина - 12) |  |
| 10 | Выход из программы | Команда 0 | Выход из программы |

# **Ответы на контрольные вопросы**

**Что такое граф?**

Граф – конечное множество вершин и соединяющих их ребер; G = <V, E>. Если пары Е (ребра) имеют направление, то граф называется ориентированным.

**Как представляются графы в памяти?**

С помощью матрицы смежности или списков смежности.

**Какие операции возможны над графами?**

Обход вершин, поиск различных путей, исключение и включение вершин.

**Какие способы обхода графов существуют?**

Обход в ширину и обход в глубину.

**Где используются графовые структуры?**

Графовые структуры могут использоваться в задачах, в которых между элементами могут быть установлены произвольные связи, необязательно иерархические.

**Какие пути в графе Вы знаете?**

Эйлеров путь, непростой путь, гамильтонов путь.

**Что такое каркасы графа?**

Каркас графа – дерево, в которое входят все вершины графа, и некоторые (необязательно все) его рёбра.

# **Вывод**

В данной лабораторной работе для хранения графа была использована матрица смежности. Для поиска недостижимых вершин в графе из заданной, подходит как алгоритм поиска в глубину, так и в ширину. Я выбрал метод поиска в глубину, так как его преимуществом является простота реализации и интуитивность алгоритма. У данного метода есть один недостаток, который заключается в том, что полученный путь, в общем случае не является кратчайшим путем из вершины v в вершину u. Данный недостаток метода никак не повлиял для нахождения недостижимых вершин.

Сам алгоритм заключается в том, что начиная с произвольной вершины v0, ищется ближайшая смежная вершина v, для которой, в свою очередь, осуществляется поиск в глубину до тех пор, пока не встретится ранее просмотренная вершина, или не закончится список смежности вершины. Если нет новых вершин, смежных с v, то вершина v считается использованной, идет возврат в вершину, из которой попали в вершину v, и процесс продолжается до тех пор, пока не получим v = v0.