|  |  |
| --- | --- |
|  | **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)** |

Отчет по лабораторной работе № 7

«Графы»

Работу выполнил:

студент группы ИУ7-32Б

Богатырев И.С.

Работу проверил:

Барышникова М.Ю.

Никульшина Т.А.

Москва, 2021 г.

**Условие задачи**

Найти все вершины графа, к которым от заданной вершины можно добраться по пути не длиннее А.

**Техническое задание**

Реализовать программу для поиска минимальных путей до каждой вершины графа, вывести результат в графической форме. Обосновать выбор использованного алгоритма.

**Входные данные:**

1. Путь до файла, содержащий данные
2. В файле:
   1. Количество узлов в графе
   2. Тройки откуда-куда-вес
3. Исходная вершина – число
4. Длина А – число.

**Выходные данные:**

1. Таблица связей
2. Перечисление подходящих вершин
3. Графическое отображение решения
4. Время, затраченное алгоритмом

**Функция программы:**

Найти расстояния до каждой вершины графа.

**Аварийные ситуации:**

1. Ввод несуществующего файла
2. Ввод пустого или содержащего некорректные данные (буквы, вещественные числа) файла
3. Некорректный ввод исходной вершины.
4. Некорректный ввод длины А.

**Структуры данных**

Для хранения связей в графе использовалась квадратная матрица целых чисел и массив типа node\_t.

typedef struct node  
*{* int dist; - расстояние от исходной вершины  
 **bool** visited; - посещение вершины  
*}* node\_t;

**Алгоритмы**

Сначала происходит чтение данных из текстового файла, по которому строится таблица связей. Потом создается массив элементов графа (node\_t \*\*), с исходными значениями dist = INT\_MAX и visited = false.

Потом запрашивается числа исходной вершины и максимальной длины.

При корректности всех введенных данных запускается алгоритм Дийкстры, который предназначен для поиска суммарного веса каждой вершины от исходной.

В конце программа выводит затраченное время и формирует digraph файл для формирования изображения.

**Тесты**

|  |  |  |
| --- | --- | --- |
| **Тест** | **Ввод** | **Вывод** |
| Несуществующий файл | test.test | Couldn't read the file |
| Пустой файл | empty.empty | Could not get the num of nodes. |
| Файл содержит не только int | Badfile.txt | Input error. |
| Файл содержит вес <= 0 | 1 2 0 | Weight couldn't be less or equal to zero |

**Вывод**

Я использовал алгоритм Дейкстры, так как он идеально подходит для данной задачи: он находит расстояния до каждой вершины, после чего остается только обработать и вывести ответ.

В моем случае сложность алгоритма составляет **O(n2)**, где n – количество вершин графа.

**Ответы на вопросы**

1. Что такое граф?

Граф – конечное множество вершин и соединяющих их ребер; G = <V, E>. Если

пары Е (ребра) имеют направление, то граф называется ориентированным; если

ребро имеет вес, то граф называется взвешенным.

2. Как представляются графы в памяти?

С помощью матрицы смежности или списков смежности.

3. Какие операции возможны над графами?

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

4. Какие способы обхода графов существуют?

Обход в ширину (BFS – Breadth First Search), обход в глубину (DFS –

Depth First Search).

5. Где используются графовые структуры?

Графовые структуры могут использоваться в задачах, в которых между

элементами могут быть установлены произвольные связи, необязательно

иерархические.

6. Какие пути в графе Вы знаете?

Эйлеров путь, простой путь, сложный путь, гамильтонов путь.

7. Что такое каркасы графа?

Каркас графа – дерево, в которое входят все вершины графа, и некоторые

(необязательно все) его рёбра.