



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 Егорова Полина Александровна

Группа ИУ7 – 34Б

Преподаватель Барышникова Марина Юрьевна

ОПИСАНИЕ УСЛОВИЯ ЗАДАЧИ

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

Указания к выполнению работы

Интерфейс программы должен быть понятен неподготовленному пользователю. При разработке интерфейса программы следует предусмотреть:

- указание формата и диапазона вводимых данных,
- блокирование ввода данных, неверных по типу,
- указание операции, производимой программой:
 - добавление элемента в стек,
 - удаление элемента из стека,
 - вычисление (обработка данных);
- наличие пояснений при выводе результата.

При тестировании программы необходимо:

- проверить правильность ввода и вывода данных (т.е. их соответствие требуемому типу и формату), обеспечить адекватную реакцию программы на неверный ввод данных;
- обеспечить вывод сообщений при отсутствии входных данных («пустой ввод»);
- проверить правильность выполнения операций;
- предусмотреть вывод сообщения при поиске несуществующих путей в графе.

ОПИСАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ

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

Входные данные:

1. Целое число, представляющее собой номер команды:

целое число в диапазоне от 0 до 6.

Выходные данные:

1. Результат выполнения определенной команды.
2. Сравнение работы алгоритма Дейкстры на различных размерах.

Функции программы:

Меню

- 1 – Загрузить граф из файла
- 2 – Ввести граф вручную
- 3 – Вывести граф
- 4 – Вывести матрицу графа
- 5 – Найти вершины, путь до которых от заданной не длиннее A
- 6 – Вывести сравнительную характеристику
- 0 – Выход

Обращение к программе:

Запуск через терминал (./app.exe)

Аварийные ситуации:

1. Некорректный ввод номера команды.

На вход: число, большее чем 6 или меньшее, чем 0.

На выход: «Неверная команда.»

ОПИСАНИЕ СТРУКТУРЫ ДАННЫХ

Граф представлен в виде матрицы смежности.

```
// матрица смежности
typedef struct
{
    int count; // количество вершин
    double **data; // матрица
} matrix_r;
```

ОПИСАНИЕ РАБОТЫ АЛГОРИТМА

Для представления графа в программе используется матрица смежности. Данные для заполнения матрицы смежности считываются из файла. Далее по полученной матрице создается файл с расширением .gv, содержащий представление графа на языке DOT. Данный файл открывается с помощью программы graphviz.

У пользователя запрашивается номер вершины, от которой производится расчет кратчайших путей до каждой другой вершины с помощью алгоритма Дейкстры. Также запрашивается максимальный путь A. Далее вершины, расстояние между которыми меньше, чем A, выводятся на экран, связи между вершинами записываются в файл с расширением .gv на языке DOT для визуализации.

ПРИМЕР РАБОТЫ ПРОГРАММЫ

Входные данные

5

0 1 2 3 0

0 0 0 2 0

2 0 0 0 3

9 0 4 0 10

0 4 0 4 0

Ключевая вершина: 3

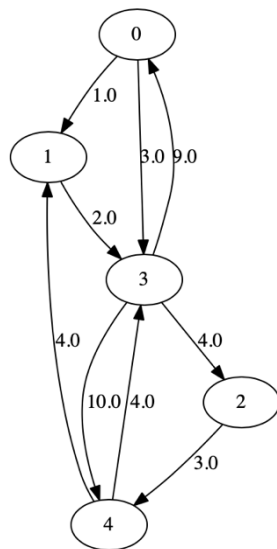
Расстояние: 10

(фиолетовые вершины – вершины, расстояние до которых меньше, чем указанное)

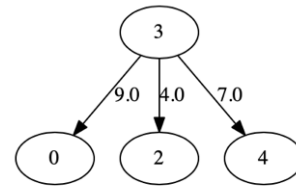
Выходные данные:

```
3 -> 0 = 9.000000
3 -> 1 = 10.000000
3 -> 2 = 4.000000
3 -> 4 = 7.000000
```

Граф по входным данным:



Граф по ключевой вершине:



НАБОР ТЕСТОВ

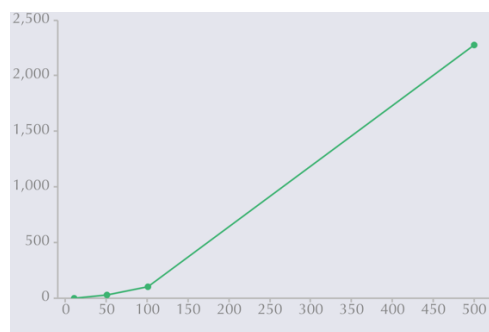
№	Название теста	Пользовательский ввод	Результат
1	Некорректный ввод команды	-12 4567 “:\$^	Неверная команда.
2	Неверный выбор номера файла (пункт 1)	Пустой ввод.	Неверно выбран файл.
3	Неверное количество вершин или вес ребра (пункт 2)	-4 Dg “*&^0%	Неверный элемент.

4	Вывод графа, вывод матрицы графа или поиск в несозданном графе (пункт 3/4/5)		Данные еще не были загружены. Выберите пункт 1.
5	Верный пункт меню		Действие выполнено.
6	Ввод максимально возможной длины, меньше, чем самый маленький путь (пункт 5)		Вывод путей и их длин. Граф не отображается.
7	Ввод максимально возможной длины, большой, чем самый большой путь (пункт 5)		Вывод путей и их длин. Граф отображается.
8	Файла, который необходимо открыть, не существует (пункт 1/4/6)		Ошибка открытия файла.

СРАВНИТЕЛЬНАЯ ХАРАКТЕРИСТИКА

Сравнивается время и память, затраченные при поиске кратчайших путей от выбранной вершины до каждой другой с помощью алгоритма Дейкстры. Время измеряется в тактах, память – в байтах.

Количество*	Время	Память
500	2286	4008
100	100	808
50	31	408
10	3	88



*Количество – количество вершин графа.

Временная сложность алгоритма – $O(n^2)$. Данный факт подтверждается экспериментально.

РАЗРАБОТАННАЯ ЗАДАЧА

Необходимо предложить вариант задачи, для которой можно было бы использовать разработанную программу.

Поскольку графы популярны при решении заданий о путях и дорогах, подобных задач придумано очень много. Поэтому мной также предложена задача в данной сфере, но несколько иного рода: в офисах Яндекса тестируются роботы-доставщики, мигрирующие между отделами. Для составления отчетов от роботов в систему поступает информация о времени, которое доставщики тратят на перемещение между отделами. Данные передаются разработанной программе, в следствие чего появляется информация о маршрутах, которые роботы проходят быстрее всего.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое граф?

Граф – это конечное множество вершин и ребер, соединяющих их, $G = \langle V, E \rangle$, где V – конечное непустое множество вершин; E – множество ребер (пар вершин).

Если пары E (ребра) имеют направление, то граф называется ориентированным (*орграф*), если иначе - неориентированный (*неорграф*). Если в пары E входят только различные вершины, то в графе нет петель. Если ребро графа имеет вес, то граф называется *взвешенным*.

Неорграф называется *связным*, если существует путь из каждой вершины в любую другую.

2. Как представляются графы в памяти?

Граф в памяти представляется в виде матрицы смежности или списка смежности.

Матрица смежности $B(n \times n)$ – элемент $b[i,j]=1$, если существует ребро, связывающее вершины i и j , и $=0$, если ребра не существует.

Список смежностей содержит для каждой вершины из множества вершин V список тех вершин, которые непосредственно связаны с ней. Входы в списки смежностей могут храниться в отдельной таблице, либо же каждая вершина может хранить свой список смежностей.

3. Какие операции возможны над графами?

- поиск кратчайшего пути от одной вершины к другой (если он есть);
- поиск кратчайшего пути от одной вершины ко всем другим;
- поиск кратчайших путей между всеми вершинами;
- поиск эйлера пути (если он есть);

- поиск гамильтонова пути (если он есть).

4. Какие способы обхода графов существуют?

Обход в ширину (BFS – Breadth First Search) - обработка вершины V осуществляется путём просмотра сразу всех «новых» соседей этой вершины, которые последовательно заносятся в очередь просмотра.

Обход в глубину (DFS – Depth First Search) - начиная с некоторой вершины v_0 , ищется ближайшая смежная ей 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 v (то есть вершина полностью обработана). 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 $v = v_0$. При просмотре используется стек.

5. Где используются графовые структуры?

Графовые структуры могут использоваться в задачах, где между элементами могут быть установлены произвольные связи.
Распространенное применение — решение задач о путях.

6. Какие пути в графе Вы знаете?

Эйлеровый путь - путь в графе, проходящий через каждое ребро ровно один раз. Если путь проходит по некоторым вершинам несколько раз — он называется *непростым*, иначе — *простым*.

Гамильтонов путь - путь, проходящий через каждую вершину ровно один раз.

7. Что такое каркасы графа?

Каркас графа – дерево, в которое входят все вершины графа, и некоторые (не обязательно все) его рёбра. Для построения каркасов графа используются алгоритмы Крускала и Прима.

Вывод

Для решения данной задачи был выбран способ хранения графа в виде матрицы смежности. Плюсы данного метода заключаются в возможности сразу визуально понять, насколько плотно связан граф, а также в легкости перевода на язык DOT связей вершин, что позволяет быстро получить графическое изображение графа.

Также для решения задачи использовался алгоритм Дейкстры – достаточно быстрый, особенно при небольших размерах графа (что было выявлено экспериментально), и простой метод нахождения кратчайших путей между вершинами. Алгоритм Дейкстры основан на выборе для включения в путь всякий раз той вершины, которая имеет наименьшую оценку кратчайшего пути (по весам ребер), то есть наименьший путь до этой вершины из всех возможных путей, которые были рассмотрены ранее.

Графы полезны при решении задач о путях или об иных связях и зависимостях. Они помогают при отображении и визуализации связей, что иногда является ключевым для быстрого решения задач.