



Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана»

ФАКУЛЬТЕТ _____ Информатика и системы управления
КАФЕДРА _____ Программное обеспечение ЭВМ и информационные технологии

Отчет по дисциплине «Типы и структуры данных» Лабораторная работа №7

Вариант 3

Студент _____ Голикова С.М.
Группа _____ ИУ7-35Б

Москва, 2021

1. Условие задачи

Реализовать алгоритмы обработки графовых структур. Найти самый длинный простой путь в графе.

2. Внешняя спецификация

Внешний вид меню:

```
Выберите действие:
1 - Прочитать матрицу смежности графа из файла
2 - Напечатать матрицу смежности
3 - Найти самый длинный простой путь в графе
4 - Создать схему графа
0 - Выйти из программы

Ваш выбор: |
```

а) Исходные данные и результаты

Входные данные:

- Целое число от 0 до 4 — номер пункта меню, который вызывает описанное в пункте действие
- Имя файла, содержащего матрицу смежности (передается в качестве аргумента командной строки)

Выходные данные:

В зависимости от пункта меню:

- Матрица смежности, считанная из файла
- Вершины самого длинного простого пути
- Графический файл

b) Задачи, реализуемые программой

- Считывание матрицы смежности из файла
- Нахождение самого длинного простого пути в графе
- Вывод графа в файл с расширением .dot

c) Способ обращения к программе

Программа запускается из терминала в директории с проектом при помощи команды «./app.exe filename.txt» (filename - имя файла, из которого происходит считывание матрицы смежности).

После создания файла с расширением .dot (пункт 4 меню) графический файл может быть создан с помощью команды “dot -Tpng graph.dot -o graph.png” в командной строке.

d) Возможные аварийные ситуации и ошибки пользователя

Аварийные ситуации и ошибки:

- некорректный выбор пункта меню
- некорректное имя файла
- пустой файл
- попытка найти самый длинный простой путь в незаполненном графе
- в графе нет простых путей (вершины не соединены между собой)

В случае аварийной ситуации пользователю выдается сообщение об ошибке и происходит возвращение в меню.

3. Внутренние структуры данных

```
// структура для хранения графа

typedef struct
{
    int size;           // размер матрицы смежности
    int **matrix;       // матрица смежности
    int path_len;       // длина самого длинного простого пути
    int *longest_path;  // самый длинный простой путь
} graph_t;
```

4. Описание алгоритма

Для нахождения самого длинного простого пути в графе ищутся самые длинные простые пути для каждой из его вершин, а потом среди этих путей ищется самый длинный.

Используется алгоритм поиска в глубину: идем “вглубь” графа, пока это возможно. Алгоритм реализуется с помощью рекурсии.

Во время данного обхода мы отмечаем посещенные вершины, а для текущей вершины записываем максимальный путь до нее (сравниваем текущий путь с длиной пути, ранее записанной в эту вершину, и выбираем большее значение).

5. Основные функции, используемые в программе

Печать матрицы смежности и самого длинного пути:

```
void print_matrix(int **matr, int size);  
void print_path(int *arr, int len);
```

Считывание матрицы смежности из файла:

```
graph_t read_adjacency_matrix_from_file(char *filename, int  
*rc);
```

Запись графа в файл с расширением .dot для дальнейшего графического представления:

```
void graph_visualization(char *filename, graph_t *graph);
```

Выделение памяти под матрицу:

```
int **allocate_matrix(int size);
```

Освобождение памяти, выделенной под поля графа:

```
void free_graph(graph_t *graph);
```

Инициализация графа:

```
void graph_init(graph_t *graph);
```

Нахождение самого длинного простого пути:

```
int longest_path(graph_t *graph);
```

Поиск в глубину:

```
void dfs(graph_t *graph, int cur_q, int *visited, int *M, int  
*q_count, int *cur_path, int *cur_path_len);
```

Проверка, что в графе нет простых путей:

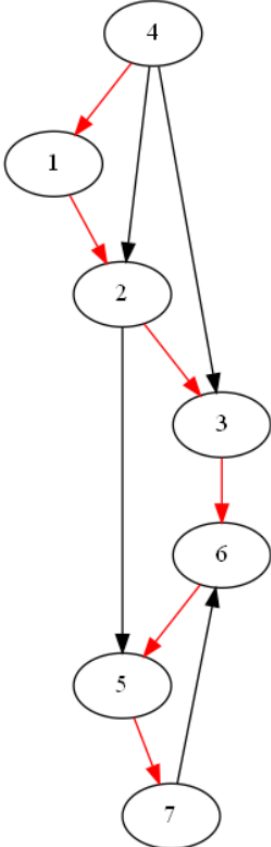
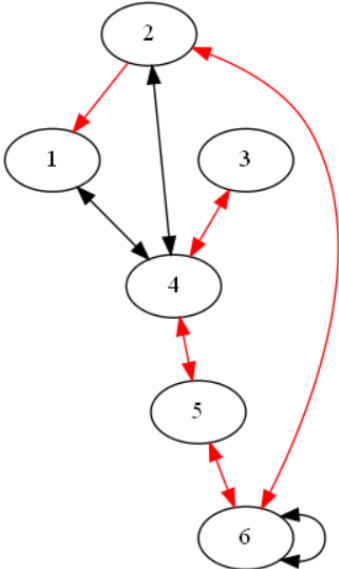
```
int no_paths(graph_t *graph);
```

6. Тесты

Негативные тесты:

№ теста	Ввод	Действие	Вывод
1	5	Ввод неверного пункта меню	Ошибка ввода: необходимо ввести номер одного из предложенных вариантов
2	1	Неверное имя файла	Ошибка при чтении матрицы смежности из файла
3	1	Пустой файл	Ошибка при чтении матрицы смежности из файла
4	2-4	Данные не загружены	Граф пуст. Выберите пункт 1, чтобы прочитать матрицу смежности графа из файла
5	3	Поиск самого длинного простого пути в графе, вершины которого не связаны	В графе нет простых путей

Позитивные тесты:

Матрица смежности	Граф с самым длинным простым путем
<p>7</p> <pre> 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 </pre>	 <p>The graph consists of 7 nodes labeled 1 through 7. The edges are as follows: 4 to 1 (red), 4 to 2 (black), 4 to 3 (black), 1 to 2 (red), 2 to 3 (red), 2 to 5 (black), 3 to 6 (red), 5 to 6 (red), 5 to 7 (red), and 6 to 7 (black). The longest simple path is highlighted in red and consists of the sequence of nodes: 1 → 2 → 3 → 6 → 5 → 7.</p>
<p>6</p> <pre> 0 0 0 1 0 0 1 0 0 1 0 1 0 0 0 1 0 0 1 1 1 0 1 0 0 0 0 1 0 1 0 1 0 0 1 1 </pre>	 <p>The graph consists of 6 nodes labeled 1 through 6. The edges are as follows: 2 to 1 (red), 2 to 4 (black), 2 to 3 (red), 1 to 4 (black), 3 to 4 (red), 4 to 5 (red), 5 to 6 (red), 6 to 2 (red), and 6 to 6 (self-loop, black). The longest simple path is highlighted in red and consists of the sequence of nodes: 1 → 4 → 5 → 6.</p>

7. Выводы

Сфера применения графов очень обширна, часто это алгоритмы поиска решений - кратчайшего пути по маршруту, эффективного расположения дорог на схеме, победной игровой стратегии и т.п. В моей задаче искался самый длинный простой путь в графе, что в жизни можно использовать для расчета худшего случая (например, максимального количества локаций в видеоигре, посещенных игроком, до перехода на новый уровень). Для обработки графов используются различные алгоритмы в зависимости от задачи, которую надо решить. Часто требуется обход графа (в глубину или в ширину). В данной задаче мы использовали поиск в глубину (временная сложность: $O(V + E)$).

8. Контрольные вопросы

1. Что такое граф?

Граф – это конечное множество вершин и ребер, соединяющих их, т. е.:

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

2. Как представляются графы в памяти?

С помощью матрицы смежности или списка смежности.

3. Какие операции возможны над графами?

Обход, поиск путей, поиск каркасов, поиск фундаментальных циклов, добавление и удаление вершин.

4. Какие способы обхода графов существуют?

Обход в глубину и обход в ширину.

5. Где используются графовые структуры?

Графовые структуры могут использоваться в задачах, в которых между элементами могут быть установлены произвольные связи (необязательно иерархические). Например, обработка картографической информации.

6. Какие пути в графе Вы знаете?

- Простой путь – путь, в котором каждая из вершин графа встречается не более одного раза.
- Контур – замкнутый путь в орграфе.
- Эйлеров путь – проходит через каждое ребро ровно один раз.
- Гамильтонов путь – проходит через каждую вершину ровно один раз.

7. Что такое каркасы графа?

Каркас графа - это связный подграф этого графа, содержащий все вершины графа и не имеющий циклов. Количество ребер в каркасе связного графа всегда на единицу меньше количества вершин графа.