

	<p>Министерство образования и науки 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)</p>
--	---

ФАКУЛЬТЕТ _____ Информатика и системы управления (ИУ)

КАФЕДРА _____ Программное обеспечение ЭВМ и информационные технологии (ИУ7)

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7

"Графы"

Студент группы ИУ7-31Б
Костев Дмитрий

Москва 2020

Цель работы

Реализовать алгоритмы обработки графовых структур: поиск различных путей, проверка связности, построение остовых деревьев минимальной стоимости.

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.

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

Техническое задание:

Входные данные:

0. Файл для построения графа

- в начале указано кол-во вершин графа
- не пустой
- вершины нумеруются последовательно, начиная с единицы
- связи между вершины указываются построчно в формате $A \rightarrow B$, где А и В номера вершин

1. Номер команды - целое число в диапазоне от 0 до 3 включительно.

2. Командно-зависимые данные:

- номер вершины

Выходные данные:

В зависимости от выбранного действия результатом работы программы могут являться:

1. Графическое представление

- орграфа
- достижимости вершины

2. Таблицы

- смежности
- достижимости

3. Информация о недостижимости вершин

Интерфейс программы:

1. Показать граф

2. Напечатать матрицы смежности и достижимости

3. Найти все вершины, недостижимые из заданной вершины

4.

Аварийные ситуации:

1. Некорректный ввод номера команды.

- **На входе:** число, большее, чем максимальный индекс команды или меньшее, чем минимальный.

- **На выходе:** Сообщение об ошибке

2. Ошибка открытия файла

- **На входе:** отсутствие файла, запрещен доступ
- **На выходе:** Сообщение об ошибке

3. Некорректный ввод номера вершины

- **На входе:** ввод, отличный от указанного в ТЗ
- **На выходе:** Сообщение об ошибке

Обращение к программе

Исполняемый файл `app.exe`. Запускается из терминала при помощи команды

`./app.exe filename`, где `filename` - имя файла с данными.

Используемые структуры

Для хранения матриц используется следующая структура.

```
typedef struct matrix {  
    size_t rows; // Кол-во строк  
    size_t cols; // Кол-во столбцов  
    int **data; // Буффер с данными  
} matrix_t;
```

Для хранения орграфа используется следующая структура

```
typedef struct graph {  
    size_t num; // Кол-во вершин графа  
    matrix_t* adjacency_matrix; // Матрица смежности  
    matrix_t* reachability; // Матрица достижимости  
} graph_t;
```

Такое хранение удобно скоростью доступа к элементам, но при этом занимает больше места.

Основные функции

`matrix_t *init_matrix(size_t rows, size_t cols);` - создание матриц

`matrix_t *sum(const matrix_t *l, const matrix_t *r);` - найти сумму матриц

`matrix_t *mul(const matrix_t *left, const matrix_t *right);` - найти произведение матриц

`matrix_t *copy_matrix(matrix_t *source);` - копирование матрицы

`free_matrix(matrix_t *matrix);` - освобождение матрицы

`graph_t* create_from_file(FILE* file);` - создание орграфа из файла

`int generate_reachability_matrix(graph_t* graph);` - генерация матрицы достижимости для графа

`void free_graph(graph_t* graph);` - освобождение графа

`int export_dot(FILE* file, graph_t* graph);` - экспорт графа в формате DOT

`int export_dot_reachability(FILE* file, graph_t* graph, size_t point);` - экспорт информации о достижимости вершин графа указанной вершиной в формате DOT

Расчет матрицы достижимости

Матрицу достижимости нахожу как сумму степеней матрицы смежности от 1 до кол-ва вершин графа.

$$B = \sum_{i=1}^n A^i$$

1. Копирование матрицы смежности дважды

- для результирующей таблицы
- для временной таблицы

2. Цикл $n - 1$ раз

- временная таблица умножается на таблицу смежности

Контрольные вопросы

1. Что такое граф?
 - Граф – конечное множество вершин и соединяющих их ребер; $G = \langle V, E \rangle$. Если пары E (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.
3. Какие операции возможны над графами?
 - Обход вершин, поиск различных путей, исключение и включение вершин.
4. Какие способы обхода графов существуют?
 - Обход в ширину обход в глубину
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.

Вывод

Один из способов проверки достижимости - это построение таблицы достижимости. Данная таблица имеет преимущество - данные о достижимости получаются сразу для всех вершин орграфа. Следовательно в задачах, связанных с постоянным поиском данная таблица упростит задачу до $O(n)$. Но данная операция очень затратна по времени построения таблицы и требуется большое кол-во памяти. Сложность выполнения операции построения таблицы смежности $O(n^3)$. Объём занимаемой памяти $O(n^2)$.

Делаем вывод, что данный способ эффективен только для маленьких связанных графов. При таком хранении больших малосвязных орграфов, очень быстро растёт время построения и объём занимаемой памяти, следовательно при таких условиях использование данной СД - нецелесообразно.