

Лабораторная работа №7. "Графы"

Студент Ларин Владимир - ИУ7-34Б

Описание условия задачи

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

Техническое задание

Входные данные:

0. Файл для построения графа
 - файл не пустой
 - в начале указано кол-во вершин графа
 - вершины нумеруются последовательно, начиная с единицы
 - связи между вершины указываются построчно в формате `A->B`, где `A` и `B` номера вершин
1. Номер команды - целое число в диапазоне от `0` до `15` включительно.
2. Командно-зависимые данные:
 - номер вершины, больше нуля, меньше или равен кол-ву вершин в орграфе

Выходные данные:

В зависимости от выбранного действия результатом работы программы могут являться:

1. Графическое представление
 - орграфа
 - достижимости той или иной вершины
2. Таблицы
 - смежности
 - достижимости
3. Информация о недостижимости вершин

Команды программы

- Перестроить граф из файла
- Показать граф

- Напечатать матрицы смежности и достижимости
- Найти все вершины, недостижимые из заданной вершины

Обращение к программе:

Запускается из терминала при помощи команды `./bin/app.out filename`, где `filename` - имя файла с данными.

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

Структуры данных

Матрицы

Для хранения матриц используется следующая структура

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

Ориентированный граф

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;
```

Описание основных функций

Матрицы

- `matrix_t *init_matrix(size_t rows, size_t cols);` - создание матрицы
- `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
- `void free_matrix(matrix_t *matrix);` - освобождение матрицы

Граф

- `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

Алгоритм

1. На экран пользователю выводится меню
2. Пользователь вводит номер команды
3. Выполняется действие согласно номеру команды

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

- к результирующей матрице добавляется временная таблица

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 $O(n^3)$. Объем занимаемой памяти $O(n^2)$.

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