

# Работа с разреженными матрицами

Алексей Сальников

## 1. Введение

Разреженные матрицы — матрицы, где большая часть элементов имеет нулевые значения. В связи с этим для работы с разреженными матрицами требуются специальные форматы хранения данных и особые алгоритмы реализации матричных операций. В большей степени это работа со списками и деревьями в противовес работе с двумерным массивом.

В рамках данного задания требуется написать несколько программ:

1. **Генератор матриц.** Матрицы и векторы записываются в текстовые файлы в определённом формате. Генераторы должны уметь генерировать объекты удовлетворяющие нужным свойствам.
2. **Умножитель.** Программа осуществляющая операцию умножения над матрицами.
3. **Построитель индекса.** Индекс — это внешняя по отношению к самой матрице структура данных, которая предназначена для ускорения работы с данными (в данном случае с матрицей). На вход построителю даётся файл с матрицей, на выходе получаем файл с представлением индекса.
4. **Редактор матрицы.** Указывается имя матрицы, позиция  $(i, j)$  в матрице и значение. В результате работы ячейка в матрице модифицируется, если индекс небыл представлен, то индекс создаётся, если был представлен, то модифицируется (запрещается создавать в этом случае индекс с нуля, надо именно модифицировать имеющийся).
5. **Отображатель индекса.** Программа, которая распечатывает индекс в понятном человеку виде<sup>1</sup>

Матрицы хранятся в виде текстовых файлов, Имя файла должно быть устроено определённым образом: В начале указывается имя матрицы, далее в качестве расширения либо **.dmtr** — для плотных матриц, либо **.smtr** — для разреженных матриц.

## 2. Организация программного кода

Желательно функции для работы с матрицами вынести в отдельный PASCAL модуль, а функции по работе с индексами (деревьями) в свой модуль. Базовую информацию по работе с модулями можно посмотреть здесь [1] и здесь [2].

В своём внутреннем представлении матрица должна храниться как бинарное дерево поиска. При этом ключём (заменой числа) в дереве должна служить пара координат  $(i, j)$ . Соответственно для любых  $i_1, i_2$  и  $j_1, j_2$  необходимо определить операцию меньше либо равно, которая даст однозначный ответ  $(i_1, j_1) \leq (i_2, j_2)$ . Можно в этом месте определить приоритет строки над столбцом (аналог лексикографического порядка в строке). Далее приведён пример элемента дерева:

---

<sup>1</sup>Поскольку индексами являются деревья, то в этом месте предполагается печать дерева.

```

type
  tree_node_t = record
    node_number : longword;
    row, column : longword;
    element      : double;
    left, right  : ^tree_node;
  end;

```

Должны присутствовать функции, которые читают матрицу из файла и записывают в файл. Должны быть функции печатающие матрицу в поток вывода в 2-х форматах: как матрица хранится в файле, и как плотная матрица (в этом представлении все нули печатаются). Тоже самое должно работать и для деревьев, образующих индекс.

## 2.1. Генератор

Генератор, в аргументах программы должен получать размерность матрицы, степень разреженности в строке матрицы<sup>2</sup> и один из режимов генерации матрицы:

1. заполненную единицами в случайных позициях,
2. со случайными значениями,
3. с единицами в случайных позициях на диагонали.

Так же в аргументах должно быть указано имя файла с матрицей, куда будут сохранены сгенерированные данные и формат матрицы (плотная или разреженная). Должен быть необязательный аргумент, который позволяет вывести создаваемую матрицу не только в файл, но ещё в поток вывода в формате плотной матрицы.

## 2.2. Умножитель

Умножителю матриц сперва в аргументах программы указывается *epsilon*, затем формат сохранения результирующей матрицы, затем имя результирующей матрицы. После этого следует список имён матриц, которые будут перемножаться. Если для какой-то матрицы в списке индекс отсутствует необходимо будет его построить перед умножением. Значение *epsilon* учитывается так, для всех промежуточных умножений если значение элемента получается по модулю меньше *epsilon*, то в матрице, в том числе промежуточном результате значение соответствующей позиции принимается равное нулю. Напоминаю здесь, что разреженная матрица нули не хранит в памяти.

## 2.3. Построитель индекса

Данная программа запрашивает у пользователя имя файла с матрицей и имя файла с индексом, затем перестраивает индекс, или строит его если индекс отсутствует. По запросу распечатывает индекс на поток вывода.

## 2.4. Отображатель индекса

На вход принимает файл с индексом, и режим печати. На выходе печатает дерево, которое задаёт индекс. Режимы:

1. Дерево печатается так: Корень, затем левое поддерево, затем правое поддерево.
2. Дерево печатается по уровням: от корня к листьям.

---

<sup>2</sup>число с плавающей точкой в интервале (0, 1] единица означает плотную матрицу (плотную диагональ, для режима генерации 1 на диагонали)

3. Дерево печатается по уровням: от листьев к корню.

Каждый элемент дерева печатается на своей строчке. Если у дерева есть поддереву, должен печататься так же его номер, либо *NULL* если поддерево отсутствует.

## 3. Форматы файлов

### 3.1. Разреженные матрицы

Формат Файла для хранения матриц разреженных матриц. В файле могут встречаться комментарии. Комментарий начинается символом '#’.

Файл для хранения матрицы начинается со слова *sparse\_matrix* далее пробельные символы, далее число строк в матрице, пробельные символы и число столбцов в матрице.

Затем следуют координаты и числа. Координаты и числа отделяются друг от друга пробельными символами. Каждое число с координатами на своей строке. Сперва в строке идёт координата номер строки в матрице, потом номер столбца, далее, само значение в матрице. В файле могут быть пустые строки и строки состоящие целиком из комментариев. Координаты нумеруются с единицы.

#### 3.1.1. Пример файла с разреженной матрицей

```
#
# This file describes
# sparse matrix
#
sparse_matrix 50000 5000

1      1      100.0
6000   2      0.85
7      1      -3.4
22     2      -12345678.00000000000005789 # very long
                                           # number
```

### 3.2. Плотные матрицы

Файл для хранения матрицы начинается со слова *dence\_matrix* далее пробельные символы, далее число строк в матрице, пробельные символы и число столбцов в матрице. Далее через пробелы и переводы строк следуют сами значения элементов. Так же как и для разреженных матриц могут встречаться комментарии и переводы строк.

#### 3.2.1. Пример файла с плотной матрицей

```
#
# This file describes
# dence atrix
#
dence_matrix 5 5

1.1 0.0 0.0 0.0 0.0
0.0 2.2 0.0 0.0 0.0
0.0 0.0 3.3 0.0 0.0
1.0 0.0 0.0 4.4 0.0
1.0 0.0 0.0 0.0 5.5
```

### 3.3. Индекс

Файл с индексами задаётся в формате ориентированного графа graphviz [3]. Для выполнения задания используется только некоторое подмножество конструкций данного формата.

Файл начинается со слова **digraph**. Далее в фигурных скобках идёт описание самого дерева. Открывающая и закрывающая скобки, должны стоять, каждая на своей строке.

Внутри сперва идёт описание вершин. Вершина задаётся своим номером, далее, после пробельных символов следует открывающая квадратная скобка, затем **label=** далее через символы пробел идёт подряд координата *i*, координата *j*, далее через **n** значение ячейки матрицы хранимое в данных координатах. Конец, задаваемый последовательностью символов **"];**. С новой строки записан следующий узел дерева.

Далее, перед указанием связей между узлами дерева, следует обязательный комментарий **//edges**. Связи задаются следующим образом.

**номер\_вершины -> номер\_левого [label="L"]; номер\_вершины -> номер\_правого [label="R"];**

Файлу в данном формате можно дать расширение **.dot** и подать на вход утилите **dot**, входящей в пакет graphviz. В результате получим отрисованное дерево.

Пример команды:

```
dot -Tpdf -o result.pdf tree.dot
```

#### 3.3.1. Пример файла с индексом

```
digraph
{
    1 [label="12_44\n0.19"];
    2 [label="1_1\n25.4"];
    3 [label="300_2\n444.6"];
    4 [label="34_12\n55.0"];

    //edges

    4 -> 1 [label="L"]; 4 -> 2 [label="R"];
    2 -> 3 [label="R"];

}
```

### Список литературы

- [1] [https://life-prog.ru/view\\_algoritmleng.php?id=117](https://life-prog.ru/view_algoritmleng.php?id=117)
- [2] [http://www.pascal.helpov.net/index/pascal\\_modules\\_programming](http://www.pascal.helpov.net/index/pascal_modules_programming)
- [3] [https://ru.wikipedia.org/wiki/DOT\\_\(%D1%8F%D0%B7%D1%8B%D0%BA\)](https://ru.wikipedia.org/wiki/DOT_(%D1%8F%D0%B7%D1%8B%D0%BA))