Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Курсовая работа**

**по курсу «Языки и методы программирования»**

**II семестр**

**Задание №7**

**«Разряженные матрицы»**

|  |  |
| --- | --- |
| **Студент:** | Ватулин Валентин |
| **Группа:** | М8О-106Б-19 |
| **Преподаватель:** | Дубинин Алексей |
| **Оценка:** |  |
| **Дата:** |  |

Москва, 2020

**Содержание**

[Введение 3](#_Toc43752797)

[Разреженная матрица 4](#_Toc43752798)

[Что это такое 4](#_Toc43752799)

[Способы хранения 4](#_Toc43752800)

[Описание программы 6](#_Toc43752801)

[Заключение 7](#_Toc43752802)

[Список литературы 8](#_Toc43752803)

# Введение

Большое количество данных в информатике может быть представимо в матричном виде. Зачастую это удобно из-за быстрого доступа к элементам, хранящимся в ячейках, но несмотря на удобство матриц, при большом размере данных возникает проблема хранения разряженных матриц в памяти, так как много памяти уходит на пустые элементы, значение которых известно. Для решения такой проблемы было придумано несколько способов хранения разреженных матриц: в виде трех векторов, в виде одного вектора и двух векторов.

# Разреженная матрица

## Что это такое

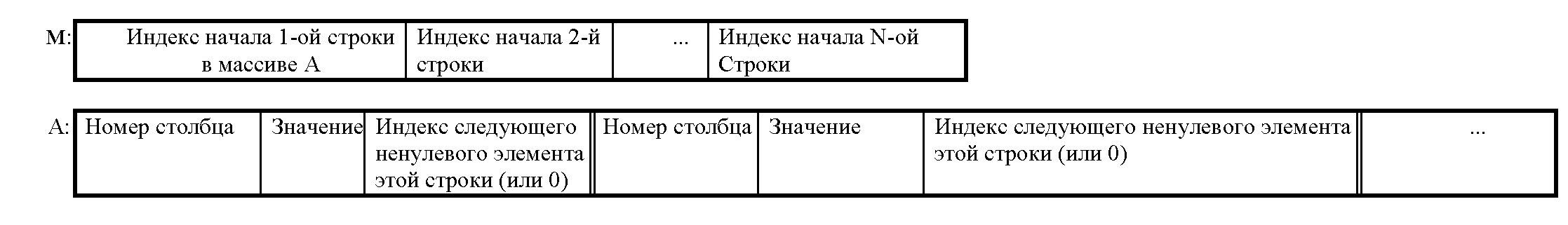
Разрежённая матрица — это матрица с преимущественно нулевыми элементами. В противном случае, если большая часть элементов матрицы ненулевые, матрица считается плотной.

При хранении и преобразовании разрежённых матриц в компьютере бывает полезно, а часто и необходимо, использовать специальные алгоритмы и структуры данных, которые учитывают разрежённую структуру матрицы. Операции и алгоритмы, применяемые для работы с обычными, плотными матрицами, применительно к большим разрежённым матрицам работают относительно медленно и требуют значительных объёмов памяти. Однако разрежённые матрицы могут быть легко сжаты путём записи только своих ненулевых элементов, что снижает требования к компьютерной памяти.

## Способы хранения

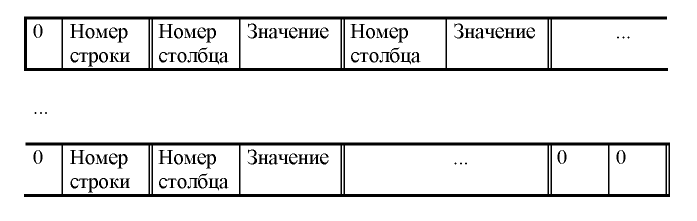
1. Цепочка ненулевых элементов в векторе А со строчным индексированием (индексы в массиве M равны 0, если соответствующая строка матрицы содержит только нули)

Индекс, равный нулю, означает отсутствие ненулевых элементов в строке (или в её остатке). Если матрицы не изменяются программой, возможна экономия памяти за счёт отказа от хранения в массиве А индексов следующего элемента столбца (когда элементы идут подряд). Вставка и удаление при этом способе возможны, но чересчур дороги: число перестановок элементов составит О(N) вместо О(1).

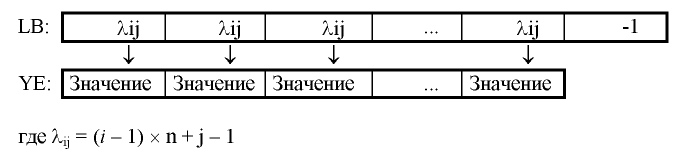


2. Один вектор:

Ненулевому элементу соответствуют две ячейки: первая содержит номер столбца, вторая содержит значение элемента. Нуль в первой ячейке означает конец строки, а вторая ячейка содержит в этом случае номер следующей хранимой строки. Нули в обеих ячейках являются признаком конца перечня ненулевых элементов разреженной матрицы.



3. Два вектора

Пусть ,

Тогда в памяти:

LB = [0,4,8]

YE=[1,1,1]

4. Связный список

В связанном списке каждый узел имеет четыре поля. Эти четыре поля определены как:

* индекс строки, где расположен ненулевой элемент
* индекс столбца, где расположен ненулевой элемент
* значение ненулевого элемента, расположенного в индексе — (строка, столбец)
* адрес следующего узла

5. Три вектора

Мы представляем исходную матрицу {\displaystyle M^{n\times m}}, cодержащую {\displaystyle N\_{NZ}} ненулевых значений в виде трёх массивов: первый вектор хранит индексы начала i строки в векторах в двух других, второй вектор хранит номер столбца, третий хранит значение.

Пусть M =

Вектор индексов: {0, 2, 3}

Вектор столбцов: {0, 1, 1, 1, 2}

Вектор значений: {1, 2, 4, 2, 6}

# Описание программы

Задание: составить программу на языке Си с процедурами и функциями для обработки прямоугольных разреженных матриц с элементами целого типа, которая выводит введенные матрицы во внутреннем представлении и в обычном виде, находит столбец, содержащий наибольшее количество ненулевых элементов, и напечатать его номер и произведение элементов этого столбца.

1) Для решения данной задачи необходимо реализовать динамический вектор.

Сложность: доступ к элементам вектора за О(1), удаление/добавление элемента за О(n).

2) Для хранения разреженной матрицы необходимо реализовать три вектора, описанных выше. Сама структура матрицы имеет следующий вид:

typedef struct {

vector val;

vector col\_ind;

vector row\_ptr;

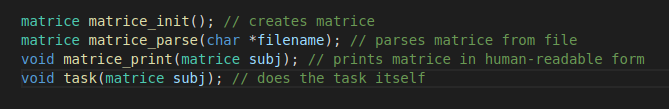
size\_t width;

} matrice;

Такое представление матрицы по памяти обойдётся в O(k), где k - количество

ненулевых элементов. Доступ к элементам векторов обходится за O(k).

3) Функции матрицы

Заключение

Эффективное по памяти хранение разреженных матриц помогает сильно уменьшить затраты программы по памяти, но при этом сильно усложняется работа с матрицами в таком представлении, так же сильно падает читаемость кода из-за того, что приходится в голове хранить несколько разных векторов в непривычном представлении. Также стоит учесть, что использование такого представление матрицы дает выигрыш лишь при большом количестве нулевых элементов.

# Список литературы

1. Разреженные матрицы

<https://studfile.net/preview/7278897/page:11/>

<https://ru.wikipedia.org/wiki/Разреженная_матрица>

2. Способы представления разреженной матрицы

<http://espressocode.top/sparse-matrix-representation/>