



Министерство науки и высшего образования 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)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

Отчет по лабораторной работе №3
«ОБРАБОТКА РАЗРЕЖЕННЫХ МАТРИЦ»

Студент Дьяченко Артём Александрович

Группа ИУ7 – 33Б

Преподаватель Рыбкин Ю. А.

Оглавление

<u>ОПИСАНИЕ УСЛОВИЯ ЗАДАЧИ.....</u>	<u>3</u>
<u>ОПИСАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ.....</u>	<u>4</u>
<u>НАБОР ТЕСТОВ.....</u>	<u>4</u>
<u>ОПИСАНИЕ СТРУКТУРЫ ДАННЫХ.....</u>	<u>5</u>
<u>ВРЕМЕННЫЕ ЗАМЕРЫ (МС).....</u>	<u>6</u>
<u>ПАМЯТЬ (БАЙТ).....</u>	<u>7</u>
<u>ОПИСАНИЕ АЛГОРИТМА.....</u>	<u>8</u>
<u>ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ.....</u>	<u>8</u>

ОПИСАНИЕ УСЛОВИЯ ЗАДАЧИ

Цель работы: реализация алгоритмов обработки разреженных матриц, сравнение этих алгоритмов со стандартными алгоритмами обработки матриц при различном размере матриц и степени их разреженности.

Разреженная (содержащая много нулей) матрица хранится в форме 3-х объектов:

- вектор A содержит значения ненулевых элементов;
- вектор JA содержит номера столбцов для элементов вектора A;
- связный список IA, в элементе N_k которого находится номер компонент в A и JA, с которых начинается описание строки N_k матрицы A.

1. Смоделировать операцию сложения двух матриц, хранящихся в этой форме, с получением результата в той же форме.
2. Произвести операцию сложения, применяя стандартный алгоритм работы с матрицами.
3. Сравнить время выполнения операций и объем памяти при использовании этих 2-х алгоритмов при различном проценте заполнения матриц.

ОПИСАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ

Входные данные:

Количество строк и столбцов матриц, количество ненулевых элементов матриц, их индексы в матрицах, вариант заполнения матриц (случайный или с клавиатуры).

Выходные данные:

Результат сложения двух матриц, сравнение двух способов обработки.

Обращение к программе:

Запускается через терминал командой: ./app.exe.

Аварийные ситуации:

1. Ввод некорректного пункта меню.
2. Некорректные значения размера матрицы.
3. Кол-во ненулевых элементов превышает кол-во элементов в матрице.
4. Размеры матриц не совпадают.

НАБОР ТЕСТОВ

№	Название теста	Пользовательский ввод	Вывод
1	Ввод матриц разной размерности	1 1 1 2 1 1 2	Размеры матриц не совпадают
2	Некорректный ввод пункта меню	3	Такого пункта нет!
3	Кол-во строк превышает макс. кол-во	10001	Некорректное количество строк
4	Кол-во ненулевых элементов превышает кол-во элементов в матрице	2 2 1 2 3 4 5	Некорректный ввод количества ненулевых элементов в матрице
5	Ввод номера строки элемента, превышающий кол-во строк в матрице	1 1 0 3 3	Номер строки превышает максимальный номер строки в матрице

6	Сложение двух матриц полной заполненности	$\begin{pmatrix} 1 \\ 2 & 2 & 4 \\ 2 & 2 & 4 \end{pmatrix}$	Первая матрица: $\begin{pmatrix} 50 & 69 \\ 78 & 36 \end{pmatrix}$ Вторая матрица: $\begin{pmatrix} 24 & 68 \\ 23 & 83 \end{pmatrix}$ Матрица результата: $\begin{pmatrix} 74 & 137 \\ 101 & 119 \end{pmatrix}$
	Сложение матрицы полной заполненности и частично заполненной матрицы	$\begin{pmatrix} 1 \\ 2 & 2 & 4 \\ 2 & 2 & 1 \end{pmatrix}$	Первая матрица: $\begin{pmatrix} 50 & 69 \\ 78 & 36 \end{pmatrix}$ Вторая матрица: $\begin{pmatrix} 0 & 0 \\ 0 & 83 \end{pmatrix}$ Матрица результата: $\begin{pmatrix} 50 & 69 \\ 78 & 119 \end{pmatrix}$

ОПИСАНИЕ СТРУКТУРЫ ДАННЫХ

```
// структура разреженной матрицы
typedef struct matrix
{
    int xsize;        // кол-во столбцов в матрице
    int ysize;        // кол-во строк в матрице
    int curr_size;    // кол-во ненулевых элементов

    int *A;           // вектор, хранящий значения ненулевых элементов
    int *JA;           // вектор, хранящий номера столбцов для элементов из A
    list_t IA;        // связный список, в элементе Nk которого находится
                        // номер компонент в A и AJ, с которых начинается
                        // описание строки Nk матрицы A
} matrix_t;

// структура матрицы в обычном виде
typedef struct std_matrix
{
    int **matrix;      // матрица элементов
    int xsize;         // кол-во столбцов в матрице
    int ysize;         // кол-во строк в матрице
} std_matrix_t;
```

```
// структура "узла" связного списка
typedef struct node
{
    int start_column_ind;    // номер компоненты в векторах A и JA
    struct node *next_element; // указатель на следующий "узел"
} node_t;

// структура связного списка IA
typedef struct list
{
    struct node *ptr; // указатель на первый "узел"
} list_t;
```

ВРЕМЕННЫЕ ЗАМЕРЫ (МС)

Заполненность, %	Размер матриц	Обычная матрица	Разряженная матрица	Отношение скоростей
47	50	27	27	1
	100	78	62	1.26
	500	807	2086	0.39
32	50	24	19	1.26
	100	33	33	1
	500	808	1589	0.51
18	50	23	12	1.92
	100	78	46	2.49
	500	827	828	1

ПАМЯТЬ (БАЙТ)

Заполненность, %	Размер матриц	Обычная матрица	Разряженная матрица	Отношение памяти
10	10	816	504	1.62
	100	80016	19224	4.16
	500	2000016	416024	4.8
50	10	816	1144	0.71
	100	80016	83224	0.96
	500	2000016	2016024	0.99
90	10	816	1784	0.46
	100	80016	147224	0.54
	500	2000016	3616024	0.55

ОПИСАНИЕ АЛГОРИТМА

1. После запуска программы пользователю предлагается или ввести две матрицы вручную, или сгенерировать их со случайными данными.
2. Пользователь указывает размерность матриц и кол-во ненулевых элементов в них.
3. Введённые матрицы хранятся в двух разных структурах: «обычной» матрице и разреженной.
4. Две матрицы складываются друг с другом, поочерёдно записывая сумму соответствующих элементов в результирующий массив, хранящийся в обоих видах. При сложении разреженных матриц поиск соответственных элементов реализуется с помощью прохода по связному списку 1А матриц.
5. В случае возникновения аварийной ситуации программа оповещает пользователя о возникшей ошибке и завершается.

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое разреженная матрица, какие способы хранения вы знаете?

Разреженная матрица — матрица, содержащая большое кол-во нулевых элементов. Хранить такую можно как обычную матрицу, с помощью линейных связных списков, кольцевых связных списков, двунаправленных стеков и очередей.

2. Каким образом и сколько памяти выделяется под хранение разреженной и обычной матрицы?

Под обычную матрицу выделяется $N * M$ ячеек памяти. А для разреженной, в моём случае, $2 * L + M$, где L — кол-во ненулевых элементов матрицы, а M — кол-во строк.

3. Каков принцип обработки разреженной матрицы?

Т.к. разреженные матрицы содержат большое кол-во нулей, то и хранятся они в таких структурах, которые «запоминают» только ненулевые элементы. Поэтому алгоритмы обработки оперируют лишь значащими данными, что даёт выигрыш по памяти и скорости по сравнению с алгоритмами обработки обычных матриц.

4. В каком случае для матриц эффективнее применять стандартные алгоритмы обработки матриц? От чего это зависит?

Чем менее разрежена матрица, тем меньший смысл использовать для её обработки алгоритмы разреженных матриц. Начиная с 50% разреженности лучше всего и по памяти, и по времени, использовать стандартные алгоритмы обработки матриц. Так, два алгоритма обработки сравниваются по скорости сложения матриц при их размере 50x50 уже на 42% заполненности, матрицы 100x100 при 32%, а 500x500 – при 18%. Т.е. чем больше берутся матрицы, тем в меньшем проценте случаев стоит использовать стандартный алгоритм.

Вывод

Использовать специальные структуры данных для (разреженных) матриц имеет смысл лишь при большом кол-ве элементов, т.к. тогда выигрыш по памяти будет существенен, и лишь при заполненности до 40% – иначе стандартные алгоритмы обработки матриц будут эффективнее во всех случаях, начиная с размерности 50x50, нежели те, которые реализуются для обработки разреженных матриц. Тем более, чем бОльшая размерность у матриц, тем меньше процент заполненности ненулевыми элементами необходим для того, чтобы стандартный способ сложения матриц превосходил по скорости способ обработки разреженных.