



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 РАЗРЯЖЕННЫХ МАТРИЦ»**

Студент Козлова Ирина Васильевна

Группа ИУ7 – 32Б

Оглавление

<u>ОПИСАНИЕ УСЛОВИЯ ЗАДАЧИ.....</u>	<u>3</u>
<u>ОПИСАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ.....</u>	<u>4</u>
<u>ОПИСАНИЕ СТРУКТУРЫ ДАННЫХ.....</u>	<u>6</u>
<u>ОПИСАНИЕ АЛГОРИТМА.....</u>	<u>8</u>
<u>НАБОР ТЕСТОВ.....</u>	<u>8</u>
<u>ОЦЕНКА ЭФФЕКТИВНОСТИ.....</u>	<u>12</u>
<u>ПАМЯТЬ.....</u>	<u>14</u>
<u>ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ.....</u>	<u>16</u>
<u>ВЫВОД.....</u>	<u>18</u>

ОПИСАНИЕ УСЛОВИЯ ЗАДАЧИ

Разработать программу умножения или сложения разреженных матриц. Предусмотреть возможность ввода данных, как с клавиатуры, так и использования заранее подготовленных данных. Матрицы хранятся и выводятся в форме трех объектов. Для небольших матриц можно дополнительно вывести матрицу в виде матрицы. Величина матриц - любая (допустим, 1000×1000). Сравнить эффективность (по памяти и по времени выполнения) стандартных алгоритмов обработки матриц с алгоритмами обработки разреженных матриц при различной степени разреженности матриц и различной размерности матриц.

Указания к выполнению работы

Все логически завершенные фрагменты алгоритма (ввод, вывод, обработка и т.п.) необходимо оформить как подпрограммы. При разработке интерфейса программы следует предусмотреть:

- указание формата и диапазона вводимых данных,
- указание операции, производимой программой,
- наличие пояснений при выводе результата,
- указание формата выводимых данных
- возможность заполнения разреженных матриц вручную (даже при большой размерности, например, 1000×1000) и автоматически с разным процентом разреженности.

При тестировании программы необходимо:

- о проверить правильность ввода о проконтролировать правильность вывода данных (т.е. их соответствие требуемому формату);
- о проверить правильность выполнения операций; о обеспечить вывод сообщений при отсутствии входных данных («пустой ввод»); о обеспечить вывод сообщений при нулевых результате или вывод нулевого результата при ненулевом входе;
- о обеспечить возможность ввода данных и вывода результата как при малых матрицах, так и при больших (например, 1000×1000).
- о сравнить время выполнения стандартного алгоритма обработки матриц и алгоритма обработки разреженных матриц при различной заполненности матриц (от 1 элемента до того количества нулей (в %), при котором становится неэффективно использование алгоритма сокращенного умножения).
- о сравнить объем требуемой памяти для реализации стандартного алгоритма обработки матриц и алгоритма обработки разреженных

матриц при различном проценте заполнения матриц и при различном их размере.

- Следует также протестировать программу при полной загрузке системы, то есть при полном заполнении матриц. Программа должна адекватно реагировать на неверный ввод, пустой ввод и выход за границы матрицы или вектора. Необходим

ОПИСАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ

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

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

1. Смоделировать операцию умножения матрицы и вектора-столбца, хранящихся в этой форме, с получением результата в той же форме.
2. Произвести операцию умножения, 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.

Входные данные:

1. **Файлы с данными:** файлы (в названии указаны размеры матриц или векторов, которые там есть) с данными для заполнения матрицы, данные хранятся в координатном виде, кол-во ненулевых элементов указывает пользователь в виде процента заполняемости.
2. **Целое число, представляющее собой номер команды:** целое число в диапазоне от 0 до 11.
3. **Командно-зависимые данные:**
 - количество строк/столбцов матрицы,
 - количество ненулевых элементов в процентах
 - индексы элементов (ненулевых) и сам элемент

Выходные данные:

1. Исходная и результирующая матрица в стандартном, разреженном или координатном виде.
2. Количественная характеристика сравнения умножения матриц разного вида.

Функции программы:

1. Заполнение матрицу из файла.
2. Заполнение матрицу вручную;
3. Заполнение вектора из файла
4. Заполнение вектора вручную;
5. Умножение классической матрицы на векторе
6. Умножение матрицы специального метода хранения на вектором
7. Вывод времени сравнения работает
8. Выходные результаты умножения в стандартном формате
9. Выходные результаты умножения в разреженном формате.
10. Выходные результаты умножения в координатном формате
11. Выведите вектор в стандартном формате.
12. Выведите вектор в разреженном виде.
13. Выведите вектор в координатном формате.
14. Выведите матрицу в стандартном форма
15. Выведите матрицу в разреженном виде.
16. Выведите матрицу в координатном формате.

Обращение к программе:

Запускается через терминал. Так же можно собрать программу используя makefile и запустить ее с помощью команды run.

Аварийные ситуации:

1. Некорректный ввод номера команды.
На входе: число, большее чем 12 или меньшее, чем 0.
На выходе: сообщение «ERROR!!! Invalid command entered, please re-enter!!!»
2. Некорректный ввод номера команды.
На входе: пустой ввод.
На выходе: сообщение «Invalid command entered, please re-enter!!!»
3. Файл пуст.

- На входе: пустой файл.
На выходе: сообщение «ERROR!!! There were problems filling in the table.»
4. Выполнение вывода матрицы/вектора до заполнения матрицы/вектора.
На входе: целое число в диапазоне от 11 до 16 (номер команды).
На выходе: сообщение «Please, input matrix/vecrot before its output.
/ERROR!!! There was no multiplication yet.»
5. Выполнение вывода умножения или количественной характеристики до умножения или до заполнения матрицы и вектора.
На входе: целое число в диапазоне от 5 до 10(номер команды).
На выходе: сообщение «Please, input matrix/vecrot before its output.
/ERROR!!! There was no multiplication yet.»
6. Неверный ввод при ручном заполнении матрицы/вектора.
На входе: строка, содержащая некорректные значения (дробь/буква).
На выходе: сообщение «Invalid row or column. ERROR!!! Problems with filling in the matrix.»
7. Неверный ввод из файла.
На входе: строка, содержащая некорректные значения (дробь/буква).
На выходе: сообщение «Invalid row or column. ERROR!!! Problems with filling in the matrix.»
8. Ввод недопустимого выбора при выборном вводе.
На входе: целое число, отличающееся от обусловленных допустимых значений для поля.
На выходе: сообщение «ERROR!!! Invalid selection entered.»
9. При умножении матриц и вектора не допустимых размеров.
На входе: матрица и вектор.
На выходе: «Invalid matrix sizes.ERROR!!! Failed to perform a multiplicati.»
10. Некорректный ввод рамеров матрицы/вектора при ручном вводе.
На входе: размер матрицы или вектора. (больше допустимых или меньше 0)
На выходе: «ERROR!!! Invalid size entered.»

ОПИСАНИЕ СТРУКТУРЫ ДАННЫХ

Стандартный вид матрицы:

```
typedef struct
{
    func_var rows;
    func_var columns;
    func_var data[MAX_MATRIX_SIZE][MAX_MATRIX_SIZE];
} matrix_std_r;
```

rows – количество строк

columns – количество столбцов

data – значения элементов (матрица всегда максимально допустимого размера)

Разреженный вид матрицы:

```
typedef struct
{
    func_var rows;
    func_var columns;
    func_var quan;
    func_var value[MAX_MATRIX_ELEMS];
    func_var row[MAX_MATRIX_ELEMS];
    list_elem pointer[MAX_MATRIX_SIZE + 1];
} matrix_r;
```

rows – количество строк

columns – количество столбцов

quan – количество ненулевых элементов

value – массив значений ненулевых элементов

row – массив номеров строк в которых расположены ненулевые элементы

pointer – массив указателей на столбцы

Связанный список

```
typedef struct list_elem
{
    func_var index;
    struct list_elem *next;
} list_elem;
```

index – значение индекса, с которого начинается описание столбца

next – указатель на следующий элемент списка

ОПИСАНИЕ АЛГОРИТМА

1. Выводится меню данной программы.
2. Пользователь вводит номер команды из предложенного меню.
3. Пока пользователь не введет 0 (выход из программы), ему будет предложено вводить номера команд и выполнять действия по выбору.

НАБОР ТЕСТОВ

	Название теста	Пользовательский ввод	Результат
1	Некорректный ввод команды	45 (допустимо от 0 до 16)	Invalid command entered, please re-enter!!!
2	Пустой ввод	Пустой ввод (нажат Enter)	Invalid command entered, please re-enter!!! / Invalid size entered./ Failed to create a result matrix / Please, input matrix/vecrot before its output. / There was no multiplication yet.
3	Некорректный ввод из файла (файл пустой)	In_02.txt	ERROR!!! There were problems filling in the table.
4	Некорректный ввод из файла (файл не существует)	In_03.txt	ERROR!!! There were problems filling in the table.

5	При вызове команды 1, 3 введена неверный выбор предложенных размеров.	6 (допустимо от 1 до 4)	ERROR!!! Invalid selection entered.
6	При вызове команды 1, 3 введен неверный процент заполнения матрицы	399 (допустимо от 1 до 100)	ERROR!!! Invalid percentage entered.
7	При вызове команды 2 вводится неверный размер матрицы	485 45 (допустимый размер до 200)	ERROR!!! Invalid size entered.
8	При вызове команды 2 вводится неверное количество ненулевых элементов	300 (при введенном размере матрицы 5x5, значит максимум количество ненулевых элементов 25)	ERROR!!! Entered the wrong number.
9	При вызове команды 2, 4 вводится неверные координаты или неверный ненулевой элемент	1 646 45 (большой номер столбца) 454 2 45 (большой номер строки) 1,15 4 8 (дробный номер столбца) 7 3,12 45 (дробный номер строки)	Invalid row or column. ERROR!!! Problems with filling in the matrix.

		<p>1 8 4,5 (дробный элемент)</p> <p>a 1 6 (буква вместо номера строки)</p> <p>5 d 5 (буква вместо номера столбца)</p> <p>58 7 fr (буква вместо элемента)</p>	
10	При вызове команды 4 вводится неверный размер вектора	480 (допустимый размер до 200)	ERROR!!! Invalid size entered.
11	При вызове команды 4 вводится неверное количество ненулевых элементов	30 (при введенном размере вектора 5, значит максимум количество ненулевых элементов 5)	ERROR!!! Entered the wrong number.
12	При вызове команды 2 все размеры, элементы и координаты введены верно	Верные данные	Матрица успешно заполнена
13	При вызове команды 4 все размеры, элементы и координаты введены верно	Верные данные	Вектор успешно заполнен
14	При вызове команды 5, 6, 7 матрица или вектор не были заполнены	Нет матрицы/вектора	ERROR!!! Failed to create a result matrix.

15	При вызове команды 8, 9, 10 матрица или вектор не были заполнены	Нет матрицы/вектора	ERROR!!! There was no multiplication yet.
16	При вызове команды 11-16 матрица или вектор не были заполнены	Нет матрицы/вектора	Please, input matrix/vecrot before its output.
17	При вызове команды 5, 6	Верные данные	Матрица умножена на вектор
18	При вызове команды 7	Верные данные	Выведена количественная характеристика результатов
19	При вызове команды 8, 9, 10 (вывод результата)	Верные данные	Вывод результата умножение вместе с матрицей и вектором. При большом размере матрицы, вывод в стандартном виде не будет произведен.
20	При вызове команды 11, 12, 13 (вывод вектора)	Верные данные	Вывод вектора. При большом размере вектора, вывод в стандартном виде не будет

			произведен.
21	При вызове команды 14, 15, 16 (вывод вектора)	Верные данные	Вывода матрицы. При большом размере вектора, вывод в стандартном виде не будет произведен.

ОЦЕНКА ЭФФЕКТИВНОСТИ

Зависимость времени от % заполнения. (вектор-столбец всегда заполняется на 100%)

(время измерялось в тактах процессора)

При записи результата было использовано среднее количество тактов, полученное за 5 измерений.

5% заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	2950	5150
50x50	23550	47140
100x100	78730	132410

10% заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	4750	5270
50x50	43330	48560
100x100	98560	138970

20% заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	5240	5260
50x50	53500	47150
100x100	135430	134700

40 % заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	5980	5190
50x50	54100	48340
100x100	165540	145400

50 % заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	6080	5880
50x50	55460	48580
100x100	228220	143320

75% заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	7250	5400

50x50	83120	45980
100x100	283848	140540

100% заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	7540	5540
50x50	100560	45470
100x100	365410	135420

ПАМЯТЬ

5% заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	60	400
50x50	1500	10000
100x100	6000	40000

10% заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	120	400
50x50	3000	10000
100x100	12000	40000

20% заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	240	400
50x50	6000	10000
100x100	24000	40000

40 % заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	480	400
50x50	12000	10000
100x100	42000	40000

50 % заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	600	400
50x50	15000	10000
100x100	60000	40000

75% заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	900	400

50x50	22500	10000
100x100	90000	40000

100% заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	1200	400
50x50	30000	10000
100x100	120000	40000

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое разреженная матрица, какие схемы хранения таких матриц Вы знаете?

Разряженная матрица — это матрица, содержащая большое количество нулей.

Схемы хранения матрицы: связанная схема хранения (с помощью линейных связанных списков), кольцевая связанная схема хранения, двунаправленные стеки и очереди, диагональная схема хранения, строчной формат, столбцовый формат.

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

Под обычную матрицу (N – количество строк, M – количество столбцов) выделяет $N \cdot M$ ячеек памяти.

Для разреженной матрицы количество ячеек памяти зависит от способа. В случае разреженного формата требуется количество ячеек в размере $K \cdot 3$ (K — количество ненулевых элементов, умножаем на 3, так как необходимо 3 списка, для хранения информации об этих элементах).

3. Каков принцип обработки разреженной матрицы?

При обработке разреженной матрицы мы работаем только с ненулевыми элементами. Тогда количество операций будет пропорционально количеству ненулевых элементов (прямая зависимость).

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

Эффективнее применять стандартные алгоритмы выгоднее при большом количестве ненулевых элементов.

Стоит отметить, что если расход памяти в программе не так важен, но важно время выполнения программы, то в случае умножения матрицы на вектор столбец лучше воспользоваться стандартным алгоритмом при большом количестве ненулевых элементов, и умножение специального (разряженного) в случае небольшого количества ненулевых элементов.

ВЫВОД

Разреженные матрицы удобнее использовать при большом количестве нулей (или при $\sim 40\%$ ненулевых элементов) в матрице, особенно при умножении на вектор-столбец, так как тратится меньший объем памяти (примерно в 2-4 раза меньше стандартного представления), а так же время выполнения программы значительно меньше (примерно в 1,5-2 раза), чем при использовании стандартного алгоритма умножения.

Но уже при $\sim 40\%$ ненулевых элементов выгоднее использовать стандартный алгоритм, так как время становится таким же или меньше, и если важна память, то так же при 40% ненулевых элементов выгоднее использовать стандартный алгоритм, так как затраты по памяти значительно меньше у обычного представления матрицы.

Хранение матрицы в разреженном виде выгодно использовать только если матрица содержит большое количество нулей, во всех остальных случаях такой вид хранения проигрывает по памяти, так как структура данного типа довольно сильно нагружена целочисленными полями, которые содержат информацию о компонентах матрицы (занимают большой объем памяти при сравнении с обычном представлении матрицы). Так же помимо большого объема занимаемой памяти, разреженные матрицы дольше обрабатываются, так как требуется много времени, чтобы составить привычный портрет матрицы.