|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **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 Есин Денис Павлович

Группа ИУ7 – 36Б

*2022 г.*

Оглавление

[Описание условия задачи 3](#__RefHeading___Toc1493_1224043242)

[Описание технического задания 4](#__RefHeading___Toc1495_1224043242)

[Описание структуры данных 6](#__RefHeading___Toc1497_1224043242)

[Описание алгоритма 8](#__RefHeading___Toc1499_1224043242)

[Набор тестов 8](#__RefHeading___Toc1501_1224043242)

[Оценка эффективности 12](#__RefHeading___Toc4279_2612490447)

[Память 14](#__RefHeading___Toc2238_2740554051)

[Ответы на контрольные вопросы 16](#__RefHeading___Toc1503_1224043242)

[Вывод 18](#__RefHeading___Toc1505_1224043242)

# **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:

* o проверить правильность ввода o проконтролировать правильность вывода данных (т.е. их соответствие требуемому формату);
* o проверить правильность выполнения операций; o обеспечить вывод сообщений при отсутствии входных данных («пустой ввод»); o обеспечить вывод сообщений при нулевых результате или вывод нулевого результата при ненулевом входе;
* o обеспечить возможность ввода данных и вывода результата как при малых матрицах, так и при больших (например, 1000 \* 1000).
* o 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).
* o 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 JA содержит номера столбцов для элементов вектора A;
* - связный список IA, в элементе Nk которого находится номер компонент в A и JA, с которых начинается описание строки 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, и степень их разреженности) с данными для заполнения матрицы, данные хранятся в координатном виде, причем индексация начинается с 0, количество ненулевых элементов указывает пользователь в виде процента заполняемости непосредственно при вызове нужного пункта в меню.
2. Целое число, представляющее собой номер команды: целое число в диапазоне от 0 до 17.
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. Выход из программы
2. Заполнение матрицы из файла
3. Ввод матрицы в консоли
4. Заполнение вектора из файла
5. Ввод вектора из консоли
6. Классическое умножение матрицы на вектор
7. Умножение матрицы на вектор, представленных в разреженном строчном формате
8. Сравнение умножения в двух различных формах
9. Вывести результат умножения в стандартном виде
10. Вывести результат умножения в разреженном строчном формате
11. Вывести результат умножения в координатном формате
12. Вывести вектор в стандартном виде
13. Вывести вектор в разреженном строчном формате
14. Вывести вектор в координатном формате
15. Вывести матрицу в стандартном виде
16. Вывести матрицу в разреженном строчном формате
17. Вывести матрицу в координатном формате
18. Вывести меню на экран

Обращение к программе:

Запускается через терминал с помощью команды ./app.exe для Unix-подобных систем и app.exe для Windows.

Аварийные ситуации:

1. Некорректный ввод номера команды.

На входе: число, большее чем 17 или меньшее, чем 0.

На выходе: сообщение «Error: unknown command!»

2. Некорректный ввод номера команды.

На входе: пустой ввод.

На выходе: сообщение «Error: command must be an integer number!»

3. Файл пуст.

На входе: пустой файл.

На выходе: сообщение «Error: read error! Check the data»

4. Выполнение вывода матрицы/вектора до заполнения матрицы/вектора.

На входе: целое число в диапазоне от 11 до 16 (номер команды).

На выходе: сообщение «Error: matrix or vector is empty 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 « Error: matrix or vector is empty yet.»

6. Неверный ввод при ручном заполнении матрицы/вектора.

На входе: строка, содержащая некорректные значения (дробь/буква).

На выходе: сообщение «Error: read error! Check the data.»

7. Неверный ввод из файла.

На входе: строка, содержащая некорректные значения (дробь/буква).

На выходе: сообщение «Error: read error! Check the data.»

8. Ввод недопустимого выбора при выборном вводе.

На входе: целое число, отличающееся от обусловленных допустимых значений для поля.

На выходе: сообщение «Error: choice error!»

9. При умножении матриц и вектора не допустимых размеров.  
На входе: матрица и вектор.  
На выходе: «Error: incorrect matrix size!»

10. Некорректный ввод рамеров матрицы/вектора при ручном вводе.  
На входе: размер матрицы или вектора. (больше допустимых или меньше 0)  
На выходе: «Error: incorrect matrix size!»

# **Описание структуры данных**

Константы, используемые в программе:

#define MAX\_MATRIX\_SIZE 150

#define MAX\_MATRIX\_ELEMS (MAX\_MATRIX\_SIZE \* MAX\_MATRIX\_SIZE)

Матрица в стандартной форме:

**typedef** **struct** matrix\_std

{

**int** rows;

**int** cols;

**int** data[MAX\_MATRIX\_SIZE][MAX\_MATRIX\_SIZE];

} **matrix\_std\_t**;

*rows – количество строк.*

*cols – количество столбцов.*

*data – двумерный массив фиксированного размера (т.к. выделяется статически на стеке), содержащий элементы матрицы типа int.*

Матрица в разреженной форме:

**typedef** **struct** matrix\_sparse

{

**int** rows;

**int** cols;

**int** quan;

**int** values[MAX\_MATRIX\_ELEMS]; // AN

**int** cols\_indexes[MAX\_MATRIX\_ELEMS]; // JA

**int** rows\_beginnings[MAX\_MATRIX\_SIZE + **1**]; // IA

} **matrix\_sparse\_t**;

*rows – количество строк.*

*cols – количество столбцов.*

*quan – количество ненулевых элементов в матрице.*

*values – массив ненулевых элементов в матрице.*

*cols\_indexes – массив индексов столбцов, в которых находятся соответствующие элементы массива values.*

*rows\_begginnings – массив индексов элементов в массиве cols\_indexes, с которых начинается описание новых строк матрицы.*

# **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 | Введена некорректная команда | Любое число <0 или >17 | Error: command must be an integer number! |
| 2 | Пустой ввод | Пустой ввод (нажат Enter) | Сообщение, содержащее информацию об ошибке |
| 3 | Файла с данными не существует | Имя несуществующего файла | Error: unable to find the file! |
| 4 | Ошибка чтения из файла | Имя файла с некорректно записанными данными | Read error! Check the data. |
| 5 | При вызове команды 1, 3 введена неверный выбор предложенных размеров или некорректный процент заполненности матрицы | В случае предложенных размеров число, не лежащее на отрезке [1; 4], в случае разреженности – или больший 100 процент, или меньший 0 | Error: incorrect choice! |
| 7 | При вызове команд 2 и 4 вводится неверный размер матрицы | Больший 200 или меньший 1 | Error: incorrect matrix size! |
| 8 | При вызове команд 2 и 4 вводится неверное количество ненулевых элементов | Число, меньшее 0 или большее площади матрицы | Error: incorrect quantity of non-zero elements! |
| 9 | При вызове команд 2 и 4 в процессе ввода матрицы что-либо введено с ошибкой, либо индексы столбца и/или строки неверные (больше максимального размера - 1 или меньше 0) или введен нулевой элемент | Неправильные данные | Read error! Check the data.  ***ИЛИ***  You entered zero! Please, enter only non-zero elements. |
| 12 | При вызове команд 2 или 4 все введено верно | Верные данные | Матрица или вектор успешно заполнены |
| 14 | При вызове команд 5 или 6 матрица или вектор не были созданы | Нет матрицы/вектора | Error: incorrect quantity of non-zero elements!  ***ИЛИ***  Error: matrix or vector is empty |
| 15 | При вызове команды 8-16 матрица или вектор не были заполнены | Нет матрицы/вектора | Error: matrix or vector is empty |
| 17 | При вызове команд 5 или 6 | Верные данные | Матрица умножена на вектор |
| 18 | При вызове команды 7 | Верные данные | Выведена количественная характеристика результатов |
| 19 | При вызове команды 8-16 | Верные данные | Вывод требуемой матрицы или вектора.  При большом размере матрицы, вывод в стандартном виде не будет произведен. |

# 

# **Оценка эффективности**

Таблица, в которой содержатся результаты измерений времени выполнения умножения и результаты замера размера матриц для каждой из форм (стандартной и строчной разреженной) для разных процентов заполнения.

Стоит отметить, что процент разреженности вектора всегда совпадает с процентом разреженности матрицы, на которую он умножается.

Время измеряется в миллисекундах, память в байтах. Каждый тест запускается 10000 раз (при выключенной оптимизации компилятора), поэтому погрешности практически нулевые.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Процент разреженности | Параметр измерения | Стандартная форма | | | | Разреженная форма | | | |
| 10х10 | 50х50 | 100х100 | 150х150 | 10х10 | 50х50 | 100х100 | 150х150 |
| 10 | Время | 0.002720 | 0.010400 | 0.035760 | 0.077120 | 0.005600 | 0.006000 | 0.007760 | 0.011120 |
| Память | 90008 | 90008 | 90008 | 90008 | 180616 | 180616 | 180616 | 180616 |
| 15 | Время | 0.002480 | 0.010400 | 0.035600 | 0.076960 | 0.005520 | 0.006400 | 0.009280 | 0.015920 |
| Память | 90008 | 90008 | 90008 | 90008 | 180616 | 180616 | 180616 | 180616 |
| 20 | Время | 0.002480 | 0.010400 | 0.035680 | 0.077040 | 0.005360 | 0.006800 | 0.010640 | 0.020640 |
| Память | 90008 | 90008 | 90008 | 90008 | 180616 | 180616 | 180616 | 180616 |
| 25 | Время | 0.002400 | 0.010320 | 0.035600 | 0.076880 | 0.005520 | 0.007200 | 0.012800 | 0.029280 |
| Память | 90008 | 90008 | 90008 | 90008 | 180616 | 180616 | 180616 | 180616 |
| 30 | Время | 0.002480 | 0.010240 | 0.035600 | 0.076880 | 0.005520 | 0.007440 | 0.015120 | 0.042400 |
| Память | 90008 | 90008 | 90008 | 90008 | 180616 | 180616 | 180616 | 180616 |
| 35 | Время | 0.002480 | 0.010400 | 0.035680 | 0.077120 | 0.005600 | 0.007840 | 0.022000 | 0.058240 |
| Память | 90008 | 90008 | 90008 | 90008 | 180616 | 180616 | 180616 | 180616 |
| 40 | Время | 0.002480 | 0.010400 | 0.035440 | 0.076880 | 0.005520 | 0.008160 | 0.026160 | 0.074480 |
| Память | 90008 | 90008 | 90008 | 90008 | 180616 | 180616 | 180616 | 180616 |
| 45 | Время | 0.002480 | 0.010320 | 0.035600 | 0.076960 | 0.005600 | 0.008560 | 0.037680 | 0.090400 |
| Память | 90008 | 90008 | 90008 | 90008 | 180616 | 180616 | 180616 | 180616 |
| 50 | Время | 0.002480 | 0.010320 | 0.035440 | 0.077040 | 0.005600 | 0.009120 | 0.043440 | 0.104800 |
| Память | 90008 | 90008 | 90008 | 90008 | 180616 | 180616 | 180616 | 180616 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Процент разреженности | Параметр измерения | Отношение | | | |
| 10х10 | 50х50 | 100х100 | 150х150 |
| 10 | Время | 0.544118 | 1.697368 | 4.535354 | 6.907143 |
| Память | 0.498339 | 0.498339 | 0.498339 | 0.498339 |
| 15 | Время | 0.457143 | 1.645570 | 3.846154 | 5.176471 |
| Память | 0.498339 | 0.498339 | 0.498339 | 0.498339 |
| 20 | Время | 0.434783 | 1.529412 | 3.353383 | 3.963115 |
| Память | 0.498339 | 0.498339 | 0.498339 | 0.498339 |
| 25 | Время | 0.449275 | 1.465909 | 2.973510 | 2.860947 |
| Память | 0.498339 | 0.498339 | 0.498339 | 0.498339 |
| 30 | Время | 0.457143 | 1.382979 | 2.556818 | 1.989712 |
| Память | 0.498339 | 0.498339 | 0.498339 | 0.498339 |
| 35 | Время | 0.426471 | 1.306122 | 1.843621 | 1.483129 |
| Память | 0.498339 | 0.498339 | 0.498339 | 0.498339 |
| 40 | Время | 0.434783 | 1.233010 | 1.522034 | 1.135673 |
| Память | 0.498339 | 0.498339 | 0.498339 | 0.498339 |
| 45 | Время | 0.455882 | 1.205607 | 1.120000 | 0.935203 |
| Память | 0.498339 | 0.498339 | 0.498339 | 0.498339 |
| 50 | Время | 0.450704 | 1.121739 | 0.973970 | 0.799835 |
| Память | 0.498339 | 0.498339 | 0.498339 | 0.498339 |

# **Ответы на контрольные вопросы**

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

Разряженная матрица — это матрица, содержащая преимущественно нулевые элементы.   
Схемы хранения матрицы: связанная схема хранения (с помощью линейных связанных списков), кольцевая связанная схема хранения, двунаправленные стеки и очереди, диагональная схема хранения, разреженный строчный и столбцовый форматы.

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

В случае стандартной матрицы в памяти выделяется rows \* cols \* size\_of\_matrix\_elem байт памяти. Память может выделяться как единым блоком, так и построчно. В последнем случае строки могут быть расположены в памяти как угодно.

В случае иных способов размер занимаемой памяти и форма хранения матрицы различаются. Так, для разреженного строчного формата выделяется два массива по общему размеру матрицы в элементах (то есть rows \* cols) и дополнительно создается список, в котором хранятся указатели на начала строк матрицы в первых двух массивах, размер списка rows + 1 элементов.

*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).

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

Для матриц эффективнее применять стандартные алгоритмы тогда, когда они или малого размера, или содержат большое количество ненулевых элементов. Это зависит, соответственно, от размера матриц и от степени их плотности.

# 

# **Вывод**

Как показывают замеры, для разреженного строчного формата в случае матрицы 10х10 вне зависимости от степени разреженности быстрее работает стандартный алгоритм, а для размеров 50х50, 100х100, 150х150 на 45-50% заполненности матрицы ненулевыми элементами разреженный строчный формат теряет свои преимущества, и далее тенденция к увеличению разрыва между скоростью работы стандартного алгоритма и нестандартного должна будет сохраняться.

Стоит отметить, что разреженный строчный формат всегда занимает приблизительно в 2 раза больше памяти, что, в принципе, объяснимо тем, что в экспериментах память под эти массивы выделялась статически на стеке (в случае выделения на куче непосредственно под требуемый размер ситуация может быть иной).

Таким образов, на малых матрицах и/или матрицах с большим количеством ненулевых элементов (>= 45% от полного размера матрицы) выгоднее использовать стандартный алгоритм, в остальных случаях предпочтительнее выбрать иной алгоритм обработки и хранения матрицы.