|  |  |
| --- | --- |
| **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 РАЗРЕЖЕННЫХ МАТРИЦ»**

Студент Ковалец Кирилл

Группа ИУ7 – 33Б

*2020 г.*

**Описание условия задачи**

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.

**Описание технического задания**

**Входные данные:**Номер команды, отвечающий за определённое действие.

**Команды:**

1. Смоделировать операцию умножения матрицы и вектора-столбца, хранящихся в особой 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.

0. Выйти из программы.

**Выходные данные:**

1. Матрица, полученная после умножения исходной матрицы на вектор столбец, в стандартной или особой форме;
2. Результат сравнения 2-х алгоритмов умножения матриц по времени и памяти.

**Обращение к программе:**Запускается через терминал командой **make run**

**Сообщения при аварийных ситуациях:**

1. Не удалось прочитать номер команды ;
2. Номер команды должен быть >= 0 и <= 3 ;
3. Не удалось прочитать метод заполнения матрицы ;
4. Метод заполнения матрицы - число от 1 до 3 ;
5. Не удалось прочитать процент заполнения матрицы ;
6. Процент заполнения матрицы - число от 1 до 100 ;
7. Не удалось прочитать кол-во строк матрицы ;
8. Превышено максимальное кол-во строк матрицы ;
9. Кол-во строк матрицы должно быть больше 0 ;
10. Не удалось прочитать кол-во столбцов матрицы ;
11. Превышено максимальное кол-во столбцов матрицы ;
12. Кол-во столбцов матрицы должно быть больше 0 ;
13. Не удалось прочитать элемент матрицы ;
14. Не удалось прочитать кол-во ненулевых элементов матрицы ;
15. Кол-во ненулевых элементов матрицы - число от 1 до (кол-во элементов матрицы) ;
16. Этот элемент уже задан ;
17. Не удалось выделить память мод матрицу ;
18. Не удалось выделить память мод массив ненулевых элементов ;
19. Не удалось выделить память мод массив, содержащий номера столбцов ненулевых элементов ;
20. Не удалось выделить память под связный список.

**Описание структуры данных**

**matrix\_t** - структура, содержащая информацию об обычной записи матрицы.

typedef struct matrix

{

int \*\*matrix;

int numb\_rows;

int numb\_columns;

} matrix\_t;

**Поля структуры:**

1) int \*\*matrix – указатель на указатель на int (матрица, в которой хранятся переменные типа int);

2) int numb\_rows – кол-во строк матрицы;

3) int numb\_columns – кол-во столбцов матрицы.

**special\_matrix\_t** - структура, содержащая информацию об особой записи матрицы.

typedef struct special\_matrix

{

int \*vector\_a;

int \*vector\_ja;

int \*linked\_list;

int numb\_rows;

int numb\_columns;

int numb\_non\_zero\_elem;

} special\_matrix\_t;

**Поля структуры:**

1) int \*vector\_a – массив, содержащий ненулевые элементы матрицы типа int;

2) int \*vector\_ja – массив, содержащий номера столбцов ненулевых элементов матрицы;

3) int \*linked\_list – связный список, содержащий номера ненулевых элементов матрицы, с которых начинается строка;

4) int numb\_rows – кол-во строк матрицы;

5) int numb\_columns – кол-во столбцов матрицы.

6) int numb\_non\_zero\_elem – кол-во ненулевых элементов матрицы.

**Описание алгоритма**

1. Выводится меню программы (каждой команде присвоен номер);
2. Пользователь вводит номер команды, который отвечает за определённое действие;
3. Ввод осуществляется до того момента, пока не будет введён 0, являющийся признаком выхода из программы.

# **Набор тестов**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Название теста** | **Входные данные** | **Результат** |
| 1 | Номер команды - число | k | Не удалось прочитать номер команды |
| 2 | Номер команды >= 0 и <=3 | 5 | Номер команды должен быть >= 0 и <= 3 |
| 3 | Кол-во строк в матрице – число | k 3 | Не удалось прочитать кол-во строк матрицы |
| 4 | Кол-во столбцов в матрице – число | 3 k | Не удалось прочитать кол-во столбцов матрицы |
| 5 | Максимальное кол-во строк в матрице - 1000 | 1001 1 | Превышено максимальное кол-во строк матрицы |
| 6 | Максимальное кол-во столбцов в матрице - 1000 | 1 1001 | Превышено максимальное кол-во столбцов матрицы |
| 7 | Кол-во строк в матрице >0 | 0 5 | Кол-во строк матрицы должно быть больше 0 |
| 8 | Кол-во столбцов в матрице >0 | 5 0 | Кол-во столбцов матрицы должно быть больше 0 |
| 9 | Метод заполнения матрицы, хранящейся в особой форме - число от 1 до 2 | 3 | Метод заполнения матрицы - число от 1 до 2 |
| 10 | Кол-во ненулевых элементов матрицы не должно превышать кол-во всех элементов матрицы | 10  (матрица 3x3) | Кол-во ненулевых элементов матрицы - число от 1 до 9 |
| 11 | Кол-во ненулевых элементов матрицы >0 | 0  (матрица 3x3) | Кол-во ненулевых элементов матрицы - число от 1 до 9 |
| 12 | Элемент матрицы - число | k 1 1  (При заполнении вручную ненулевых элементов матрицы) | Не удалось прочитать элемент матрицы |
| 13 | Номер строки матрицы - число | 1 k 1  (При заполнении вручную ненулевых элементов матрицы) | Не удалось прочитать номер строки элемента матрицы |
| 14 | Номер столбца матрицы – число | 1 1 k  (При заполнении вручную ненулевых элементов матрицы) | Не удалось прочитать номер столбца элемента матрицы |
| 15 | Номер строки элемента матрицы не должен превышать кол-во строк в ней и быть >0 | 1 0 1  (матрица 3x3) | Номер строки матрицы - число от 1 до 3 |
| 16 | Номер столбца элемента матрицы не должен превышать кол-во столбцов в ней и быть >0 | 1 1 4  (матрица 3x3) | Номер столбца матрицы - число от 1 до 3 |
| 17 | Процент заполнения матрицы - число от 1 до 100 | 200  (При автозаполнении матрицы) | Процент заполнения матрицы - число от 1 до 100 |
| 18 | Метод заполнения обычной матрицы - число от 1 до 3 | 4 | Метод заполнения матрицы - число от 1 до 3 |
| 19 | Перемножение обычной матрицы на вектор столбец  (ввод всех элементов) | 1 0 0  0 1 0  0 0 1  1  2  3 | 1  2  3 |
| 20 | Перемножение обычной матрицы (3x3) на вектор столбец (3x1)  (ввод только ненулевых элементов) | 2 1 1  2 2 2  2 3 3  5 2 1  7 3 1 | 0  10  14 |
| 21 | Перемножение обычной матрицы на вектор столбец  (автозаполнение) | Размер матрицы – 3x3  Процент заполнения – 50% | 0 -5 3  0 0 0  -3 3 0  0  -3  0  15  0  -9 |
| 22 | Перемножение особой матрицы (3х3 с 5-ю ненулевыми элементами) на вектор столбец (3х1 с 3 ненулевыми элементами)  (заполнение вручную) | 1 1 1  1 1 2  1 1 3  2 2 2  2 3 3  1 1 1  2 2 1  3 3 1 | Массив из ненулевых элементов матрицы:  6 4 6  Массив из номеров столбцов ненулевых элементов матрицы:  1 1 1  Связный список:  1 2 3 4 |
| 23 | Перемножение особой матрицы на вектор столбец (автозаполнение) | Размер матрицы – 3x3  Процент заполнения – 50% | Исходная матрица:  3 -1 -3 5  1 2 3 3  1 3 4 5  Вектор:  3  1  1 1 2 2  Результат:  -3  1  1 2 2 2 |
| 24 | Вывести результаты сравнения времени выполнения операций и объёма памяти при использовании этих 2-х алгоритмов при различном проценте заполнения матриц. | Команда 3 | Вывод результатов сравнения двух алгоритмов перемножения матриц |
| 25 | Выход из программы | Команда 0 | Выход из программы |

**Оценка эффективности**

Сортировка каждой таблицы будет измеряться в тактах процессора (процес- сор со средней частотой 2.3gHz).

Матрица и вектор столбец будут заполнены случайными числами.

**Сравнение времени умножения**

**При 10% заполнения**

|  |  |  |
| --- | --- | --- |
| Размер | Матрица в особой форме | Обычная матрица |
| 10x10 | 1946 | 2266 |
| 50x50 | 3879 | 21961 |
| 100x100 | 10448 | 69456 |

**При 25% заполнения**

|  |  |  |
| --- | --- | --- |
| Размер | Матрица в особой форме | Обычная матрица |
| 10x10 | 1036 | 777 |
| 50x50 | 6573 | 18035 |
| 100x100 | 22799 | 71959 |

**При 50% заполнения**

|  |  |  |
| --- | --- | --- |
| Размер | Матрица в особой форме | Обычная матрица |
| 10x10 | 1427 | 889 |
| 50x50 | 13395 | 15699 |
| 100x100 | 40190 | 68870 |

**При 75 % заполнения**

|  |  |  |
| --- | --- | --- |
| Размер | Матрица в особой форме | Обычная матрица |
| 10x10 | 1598 | 888 |
| 50x50 | 21870 | 18346 |
| 100x100 | 65639 | 78630 |

**При 100 % заполнения**

|  |  |  |
| --- | --- | --- |
| Размер | Матрица в особой форме | Обычная матрица |
| 10x10 | 1683 | 799 |
| 50x50 | 20156 | 18516 |
| 100x100 | 77333 | 67962 |

**Сравнение затрат на память под исходную матрицу (в байтах)**

**При 10% заполнения**

|  |  |  |
| --- | --- | --- |
| Размер | Матрица в особой форме | Обычная матрица |
| 10x10 | 124 | 400 |
| 50x50 | 2204 | 10000 |
| 100x100 | 8404 | 40000 |

**При 25% заполнения**

|  |  |  |
| --- | --- | --- |
| Размер | Матрица в особой форме | Обычная матрица |
| 10x10 | 244 | 400 |
| 50x50 | 5204 | 10000 |
| 100x100 | 20404 | 40000 |

**При 50% заполнения**

|  |  |  |
| --- | --- | --- |
| Размер | Матрица в особой форме | Обычная матрица |
| 10x10 | 444 | 400 |
| 50x50 | 10204 | 10000 |
| 100x100 | 40404 | 40000 |

**При 75 % заполнения**

|  |  |  |
| --- | --- | --- |
| Размер | Матрица в особой форме | Обычная матрица |
| 10x10 | 644 | 400 |
| 50x50 | 15204 | 10000 |
| 100x100 | 60404 | 40000 |

**При 100 % заполнения**

|  |  |  |
| --- | --- | --- |
| Размер | Матрица в особой форме | Обычная матрица |
| 10x10 | 844 | 400 |
| 50x50 | 20204 | 10000 |
| 100x100 | 80404 | 40000 |

# **Ответы на контрольные вопросы**

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

Разреженная матрица — это матрица, содержащая большое количество нулей.

Схемы хранения матрицы:

- связанная схема хранения;

- 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 выделяется память (в байтах), равная кол-во строк матрицы, умноженному на кол-во её столбцов, а также на размер типа данных её элементов.

Для разреженной матрицы количество байт памяти завит от способа её хранения. В случае хранения её в особой форме необходимо три списка. Размеры двух вычисляются как произведение кол-ва ненулевых элементов матрицы на размер типа данных её элементов. Размер третьего – как произведение кол-ва строк матрицы + 1 на размер типа данных её элементов.

3. Каков принцип обработки разреженной матрицы?

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

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

Эффективнее применять стандартные алгоритмы при большом количестве ненулевых элементов. А если заранее известно, что ненулевых элементов в матрице немного относительно их общего количества, то лучше применять способы работы с разреженными матрицами.

# **Вывод**

Алгоритмы работы с разреженной матрицей эффективны только в том случае, если количество ненулывых элементов в матрице мало. Это хорошо видно при больших размерах матрицы (100х100). В этом случае только при заполнении свыше 75% алгоритм для работы с разреженной матрицей работает дольше обычного. Если взять матрицу малого размера (10х10), то особый алгоритм медленнее уже при 25% заполнения матрицы. Выходит, что для быстроты алгоритма работы с разряженными матрицами кроме кол-ва ненулевых элементов матрицы также важен её размер.

Затраты по памяти алгоритма для разряженной матрицы превышают затраты обычного алгоритма при 50% заполнения матрицы, а при 100% превосходят уже в два раза.

Алгоритм для работы с разряженными матрицами имеет свои преимущества и недостатки в сравнении с обычным. Каждый из них подходит для своих задач.