|  |  |
| --- | --- |
| **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 Лазутин Александр Владимирович

Группа ИУ7 – 33Б

Проверяющий Барышникова Марина Юрьевна

*2022 г.*

# **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.

# **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:

Целое число, представляющее собой вариант заполнения матрицы:

целое число в диапазоне от 1 (собственноручное введение матрицы) до 2 (случайное заполнение матрицы)

Дополнительный ввод: поле типа int в зависимости от требования

Выходные данные:

1. Получившаяся матрица (в обычном представлении или разреженном виде) и вектор столбец

2. Вектор, являющийся результатом перемножения (в обычном и разреженном виде)

3. Результаты замеров времени и памяти для перемножения обычного и разреженного

(пункты 1 и 2 выводятся на экран в зависимости от их размера — если много чисел, то будет выведено соответствующее сообщение)

Функции программы:

1. Самостоятельно ввести ненулевые элементы матрицы и вектора столбца, вставив их в нужную позицию
2. Сгенерировать матрицу и вектор столбец случайным образом

Обращение к программе:

Запускается через терминал командой ./app.exe

Аварийные ситуации:

1. Неверно введён способ заполнения матрицы

Входные данные: не число или число, большее 2 или меньшее 1

Выходные данные: сообщение «Ошибка: тип заполнения матрицы введён неверно»

2. Неверно введены размеры матрицы

Входные данные: не число или число, большее 10000 или меньшее 1

Выходные данные: сообщение «Ошибка: неверно введены размеры матрицы»

3. Ошибки выделения памяти

Входные данные: попытка выделить память под матрицу

Выходные данные: сообщение «Ошибка: память под матрицу выделит не удалось»

4. Неверно введено количество ненулевых элементов в матрице

Входные данные: не число или число, большее общего количества элементов в матрице или меньшее 0

Выходные данные: сообщение «Ошибка: количество ненулевых элементов в матрице введено неверно»

5. Неверно введено количество ненулевых элементов в векторе столбце

Входные данные: не число или число, большее общего количества элементов в векторе столбце или меньшее 0

Выходные данные: сообщение «Ошибка: количество ненулевых элементов в векторе столбце введено неверно»

6. Неверно введен элемент матрицы при заполнении

Входные данные: не число

Выходные данные: сообщение «Ошибка: неверно введено число»

7. Неверно введен номер строки матрицы

Входные данные: не число или число, меньшее 0 или большее количества строк матрица

Выходные данные: сообщение «Ошибка: неверно введена строка матрицы»

8. Неверно введен номер столбца матрицы

Входные данные: не число или число, меньшее 0 или большее количества столбцов матрицы

Выходные данные: сообщение «Ошибка: неверно введен столбец матрицы»

Описание структуры данных

*Структура для хранения обычной матрицы*

typedef struct

{

int \*\*matrix;

int x\_len;

int y\_len;

int no\_zeroes;

} statix\_matrix\_t;

Поля структуры:

1. int \*\*matrix — двумерный массив типа int, в котором будут храниться все элементы

2. int x\_len — количество строк матрицы

3. int y\_len — количество столбцов матрицы

4. int no\_zeroes — количество ненулевых элементов матрицы

*Структура для хранения разреженной матрицы*

typedef struct

{

int \*a;

int \*ja;

int \*ia;

int a\_len;

int ia\_len;

} matrix\_t;

Поля структуры:

1. int \*a — массив ненулевых элементов

2. int \*ja — массив столбцов ненулевых элементов

3. int \*ia — массив первых вхождений элементов в строку в матрицы

4. int a\_len — длина массива ненулевых элементов

5. int ia\_len — длина массива первых вхождений

list\_t — структура, которая хранит массив значений для вектора столбца (int \*arr) и его длину (int len)

typedef struct

{

int \*arr;

int len;

} list\_t;

# **Описание алгоритма**

1. Выводится выбор типа генерации матрицы
2. Пользователь вводит пункт, который отвечает за случайную генерацию или собственноручный ввод матрицы и вектора столбца
3. Далее программа выводит матрицу и вектор столбец, результат умножения (в обычном и разреженном виде), а также замеры времени и памяти для каждого из способов умножения

# **Набор тестов**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Название теста** | **Пользовательский ввод** | **Результат** |
| 1 | Некорректный ввод типа заполнения матрицы | аааааа | Ошибка: тип заполнения матрицы введен неверно |
| 2 | Некорректный ввод типа заполнения матрицы | 15 | Ошибка: тип заполнения матрицы введен неверно |
| 3 | Неверно введены размеры матрицы  (первое значение — не число, второе число) | а 1 | Ошибка: неверно введены размеры матрицы |
| 4 | Неверно введены размеры матрицы  (первое значение — число, второе не число) | 10 а | Ошибка: неверно введены размеры матрицы |
| 5 | Неверно введены размеры матрицы  (первое значение — число (меньше 1 или больше 10000), второе — нормальное число) | -1 10 | Ошибка: неверно введены размеры матрицы |
| 6 | Неверно введены размеры матрицы  (первое значение — нормальное число, второе — число (меньше 1 или больше 10000)) | 5 0 | Ошибка: неверно введены размеры матрицы |
| 7 | Неверно введено количество ненулевых элементов в матрице  (не число) | аааааа | Ошибка: количество ненулевых элементов в матрице введено неверно |
| 8 | Неверно введено количество ненулевых элементов в матрице  (меньше 0 или больше всего количества элементов в матрице) | -10 | Ошибка: количество ненулевых элементов в матрице введено неверно |
| 9 | Неверно введено количество ненулевых элементов в векторе столбце  (не число) | аааааа | Ошибка: количество ненулевых элементов в векторе столбце введено неверно |
| 10 | Неверно введено количество ненулевых элементов в векторе столбце  (меньше 0 или больше всего количества элементов в векторе столбце) | 7  (если вектор длины 6) | Ошибка: количество ненулевых элементов в векторе столбце введено неверно |
| 11 | Неверно введен элемент матрицы при заполнении  (не число) | аааааа | Ошибка: неверно введено число |
| 12 | Неверно введен номер строки матрицы  (не число) | ааааааа | Ошибка: неверно введена строка матрицы |
| 13 | Неверно введен номер строки матрицы  (меньше 0 или больше количества строк матрицы) | 10  (если строк матрицы 2) | Ошибка: неверно введена строка матрицы |
| 14 | Неверно введен номер столбца матрицы  (не число) | ааааааа | Ошибка: неверно введен столбец матрицы |
| 15 | Неверно введен номер столбца матрицы  (меньше 0 или больше количества столбцов в матрице) | -10 | Ошибка: неверно введен столбец матрицы |
| 16 | Собственноручный безошибочный ввод матрицы | 1 | Матрица (обычная и в разреженном виде), вектор столбец, результат сложения (в обычном и разреженном виде) и замеры времени и памяти |
| 17 | Случайная генерация матрицы | 2 | Матрица (обычная и в разреженном виде), вектор столбец, результат сложения (в обычном и разреженном виде) и замеры времени и памяти |
| 18 | Количество строк в матрице больше 30 | Размеры матрицы:  31 5 | Матрица не выведена из ее большого размера  все остальное выведено |
| 19 | Количество столбцов в матрице больше 30 | Размеры матрицы:  5 31 | Матрица не выведена из ее большого размера  все остальное выведено |
| 20 | Количество ненулевых элементов в матрице больше 30 | Ненулевые элементы матрицы  31 | Не выведена матрица в разреженном виде из-за ее большого размера  все остальное выведено |
| 21 | Количество ненулевых элементов в векторе столбце больше 30 | Ненулевые элементы вектора столбца  31 | Не выведен вектор столбец из-за его большого размера  все остальное выведено |
| 22 | Количество ненулевых элементов в результирующем векторе больше 35 | Результат (ненулевых элементов)  36 | Вектор результата в разреженном виде не выведен из-за его большого размера |

# **Оценка эффективности**

Сортировка каждой таблицы будет измеряться в тактах процессора на процессоре с частотой 35000000 Гц

Матрица и вектор будут заполнены случайными числами, вектор столбец будет заполнен весь.

Чтобы получить среднее время, я делал замеры каждого алгоритма 10 раз.

**Замеры времени умножения:**

10% заполнения

|  |  |  |
| --- | --- | --- |
| Размер | Разряженная матрица | Обычная матрица |
| 10x10 | 2824 | 3734 |
| 50x50 | 17524 | 68052 |
| 100x100 | 51332 | 277840 |

20% заполнения

|  |  |  |
| --- | --- | --- |
| Размер | Разряженная матрица | Обычная матрица |
| 10x10 | 3452 | 3536 |
| 50x50 | 36456 | 66672 |
| 100x100 | 85104 | 275884 |

40 % заполнения

|  |  |  |
| --- | --- | --- |
| Размер | Разряженная матрица | Обычная матрица |
| 10x10 | 3888 | 3616 |
| 50x50 | 53204 | 67508 |
| 100x100 | 137618 | 247612 |

50 % заполнения

|  |  |  |
| --- | --- | --- |
| Размер | Разряженная матрица | Обычная матрица |
| 10x10 | 4016 | 3270 |
| 50x50 | 62060 | 66016 |
| 100x100 | 166154 | 248312 |

60 % заполнения

|  |  |  |
| --- | --- | --- |
| Размер | Разряженная матрица | Обычная матрица |
| 10x10 | 4593 | 3324 |
| 50x50 | 79860 | 66132 |
| 100x100 | 288780 | 257573 |

100% заполнения

|  |  |  |
| --- | --- | --- |
| Размер | Разряженная матрица | Обычная матрица |
| 10x10 | 6120 | 3648 |
| 50x50 | 109816 | 66084 |
| 100x100 | 316320 | 246856 |

**Замеры памяти**

10% заполнения

|  |  |  |
| --- | --- | --- |
| Размер | Разряженная матрица  (байт) | Обычная матрица  (байт) |
| 10x10 | 120 | 400 |
| 50x50 | 2200 | 10000 |
| 100x100 | 8400 | 40000 |

20% заполнения

|  |  |  |
| --- | --- | --- |
| Размер | Разряженная матрица  (байт) | Обычная матрица  (байт) |
| 10x10 | 200 | 400 |
| 50x50 | 4200 | 10000 |
| 100x100 | 16400 | 40000 |

40 % заполнения

|  |  |  |
| --- | --- | --- |
| Размер | Разряженная матрица  (байт) | Обычная матрица  (байт) |
| 10x10 | 360 | 400 |
| 50x50 | 8200 | 10000 |
| 100x100 | 32400 | 40000 |

50 % заполнения

|  |  |  |
| --- | --- | --- |
| Размер | Разряженная матрица  (байт) | Обычная матрица  (байт) |
| 10x10 | 440 | 400 |
| 50x50 | 10200 | 10000 |
| 100x100 | 40400 | 40000 |

60 % заполнения

|  |  |  |
| --- | --- | --- |
| Размер | Разряженная матрица  (байт) | Обычная матрица  (байт) |
| 10x10 | 520 | 400 |
| 50x50 | 1220 | 10000 |
| 100x100 | 48400 | 40000 |

100% заполнения

|  |  |  |
| --- | --- | --- |
| Размер | Разряженная матрица  (байт) | Обычная матрица  (байт) |
| 10x10 | 840 | 400 |
| 50x50 | 20200 | 10000 |
| 100x100 | 80400 | 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, 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 память выделяется размером в байтах, равная количеству строк умноженных на количество столбцов, а также на тип данны  
Для разреженной матрицы количество ячеек памяти завит от способа

В случае разряженного формата необходимо два списка, размер каждого из которых равен количеству ненулевых элементов, помноженных на тип данных, а также список для хранения элементов первых вхождений в строки(столбцы), равный количеству строк (столбцов), помноженных на тип данных

*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 для работы с разреженной матрицой имеют смысл только, если количество ненулывых элементов в обычной матрице не много.

Начиная с заполненности 40% обычный алгоритм начинает работать быстрее разреженного для матрицы 10 Х 10. А для матриц 50 Х 50 и 100 Х 100 начиная с заполненности около 60% алгоритм для работы с разряженными матрицами начинает работать медленнее алгоритма для обычных матриц.

При заполненности уже свыше 50 процентов память, которая выделяется под хранение рареженной матрицы начинает превышать память, необходимую для хранения обычной матрицы.