



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

Цветков Иван Алексеевич

Группа

ИУ7 – 33Б

2020 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,  $1000 \times 1000$ ). 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:**

#### **1. Целое число, представляющее собой вариант заполнения матрицы:**

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

### **3. 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 выделить 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	iu	Ошибка: тип заполнения матрицы введен неверно
2	Некорректный ввод типа	15	Ошибка: тип заполнения

	заполнения матрицы		матрицы введен неверно
3	Неверно введены размеры матрицы (первое значение — не число, второе число)	Iu 23	Ошибка: неверно введены размеры матрицы
4	Неверно введены размеры матрицы (первое значение — число, второе не число)	50 iu	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))	15 0	Ошибка: неверно введены размеры матрицы
7	Неверно введено количество		Ошибка: количество

	ненулевых элементов в матрице (не число)	iu	ненулевых элементов в матрице введено неверно
8	Неверно введено количество ненулевых элементов в матрице (меньше 0 или больше всего количества элементов в матрице)	-1 или 25 (матрица 5 на 5)	Ошибка: количество ненулевых элементов в матрице введено неверно
9	Неверно введено количество ненулевых элементов в векторе столбце (не число)	iu	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)	-1 или 7 (если вектор длины 6)	Ошибка: количество ненулевых элементов в векторе столбце введено неверно



11	Неверно введен элемент матрицы при заполнении (не число)	iu	Ошибка: неверно введено число
12	Неверно введен номер строки матрицы (не число)	iu	Ошибка: неверно введена строка матрицы
13	Неверно введен номер строки матрицы (меньше 0 или больше количества строк матрицы)	-1 или 7 (если строк матрицы 6)	Ошибка: неверно введена строка матрицы
14	Неверно введен номер столбца матрицы (не число)	iu	Ошибка: неверно введен столбец матрицы
15	Неверно введен номер столбца матрицы (меньше 0 или больше количества столбцов в матрице)	-1 или 10 (если столбцов в матрице 9)	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 10	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	Ненулевые элементы	Не выведен вектор

	ненулевых элементов в векторе столбца больше 30	вектора столбца 31	столбец из-за его большого размера  все остальное выведено
22	Количество ненулевых элементов в результатирующем векторе больше 35	Результат (ненулевых элементов) 36	Вектор результата в разреженном виде не выведен из-за его большого размера
23	Количество элементов в результатирующем векторе больше 30	Результат (чисел)	Вектор результата в обычном 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

Время умножения:

5% заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	1992	3476
50x50	13448	67196
100x100	33992	285540

10% заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	2784	3848
50x50	17524	68052
100x100	51332	277840

20% заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	3452	3536
50x50	26456	66672
100x100	85104	275884

40 % заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	3888	3616
50x50	43204	67508
100x100	137618	247612

50 % заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	4016	3270
50x50	52060	66016
100x100	166154	248312

75% заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	4996	3640
50x50	67220	75612
100x100	270772	276904

100% заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	5120	3648

50x50	89816	66084
100x100	316320	246856

### **Замеры памяти**

5% заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	80	400
50x50	1200	10000
100x100	4400	40000

10% заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	120	400
50x50	2200	10000
100x100	8400	40000

20% заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	200	400
50x50	4200	10000
100x100	16400	40000

40 % заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	360	400
50x50	8200	10000
100x100	32400	40000

50 % заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	440	400
50x50	10200	10000
100x100	40400	40000

75% заполнения

Размер	Разряженная матрица	Обычная матрица
10x10	640	400
50x50	15200	10000
100x100	60400	40000

100% заполнения

Размер	Разряженная матрица	Обычная матрица
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 зависит 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 матрицей имеют смысл только, если количество ненулевых элементов в обычной матрице не много (в зависимости от алгоритма, в данном случае только при заполненности свыше 80 процентов обычный алгоритм быстрее разреженного)

Также стоит учитывать, что при заполненности уже свыше 50 процентов память, которая выделяется под хранение разреженной матрицы начинает превышать память, необходимую для хранения обычной матрицы (при больших заполненностях матрицы свыше 90 процентов требуется в 2 раза больше памяти, чем для обычной)

В итоге, стоит учитывать все обстоятельства, чтобы можно было бы наиболее эффективно пользоваться такими ресурсами, как память и время