

	<p><b>Министерство науки и высшего образования Российской Федерации</b>  <b>Федеральное государственное бюджетное образовательное учреждение</b>  <b>высшего образования</b></p> <p><b>«Московский государственный технический университет</b>  <b>имени Н.Э. Баумана</b></p>
---	---

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии

## **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3**

### **«Обработка разреженных матриц»**

Студент Фролов Евгений

Группа ИУ7 – 35Б

2020 г.

**Цель работы:** реализация алгоритмов обработки разреженных матриц, сравнение этих алгоритмов со стандартными алгоритмами обработки матриц при различном размере матриц и степени их разреженности.

**Условие задачи (3 вариант):**

Разреженная 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:**

Матрица и вектор столбец(одномерный массив). У пользователя есть возможность ввести матрицу автоматически, указывая только размеры матрицы и процент заполняемости. Тоже самое для массива; ручной ввод; ввод только ненулевых элементов и их индексы в матрице.

**Выходные данные:**

Матрица – результат умножения матрицы и вектора столбца;  
время, затраченное на умножение.

**Возможные ошибки:**

- Ввод некорректных данных
- Ввод по неправильному индексу элемента

Обращение к программе: через консоль командой ./main.exe

**Флаги при компиляции:**

```
gcc -std=c99 main.c func.c -o main.exe
```

## Описание функций:

- Функция для умножение матрицы хранящейся в связанном списке и вектора столбца  
(void multiply(int n, int \*A, int \*JA, int \*IA, int \*x, int \*b))
- Функция для ручного ввода матриц. Принимает массив и указатели на размеры матрицы.  
(int mtr\_input(int a[][NMAX], int \*n, int \*m))
- Функция для обычного умножения матрицы.  
(void multiply\_basic(int a[][NMAX], int n, int m, int \*b, int c[][NMAX]))
- Вывод массива.  
(void print\_array(int \*mtr, int len))
- Функция для преобразования матрицы в связанный список.  
(void convert\_matrix(const int matr[][NMAX], int n, int m, int \*A, int \*JA, int \*IA, int \*count))
- Функция для генерирования матрицы с определенным процентом кол-ва целых.  
(void generate\_matrix(int matr[][NMAX], int n, int m, int fill))
- Функция для вывода времени в тиках.  
(unsigned long long tick(void))
- Функция для генерирования вектора столбца с определенным кол-вом целых.  
(void generate\_array(int \*x, int n, int fill))

## Тестирование:

Входные данные	Вывод
Неверный ввод пункта меню для ввода матрицы	Input ERROR(you can choose 1, 2 , 3 only)
Неверный ввод размеров матрицы (Например: n e)	Input ERROR
Неверный ввод элементов матрицы или вектора столбца	Input ERROR
Неверный ввод кол-ва ненулевых элементов(больше чем количество элементов в матрице или символ)	Input ERROR
Неверный ввод значения, позиций в матрице ненулевых элементов	Input ERROR

## Структуры данных:

```
int a[NMAX][NMAX], n_a, m_a; //матрица, которую вводит пользователь
int c1[NMAX][NMAX]; //результатирующая матрица
int a1[NMAX], count; //массив для хранения ненулевых элементов
int ja1[NMAX]; // массив для хранения индексов столбцов
int ia1[NMAX]; // связный список IA, в элементе Nk которого находится номер
компонент в A и JA, с которых начинается описание строки Nk матрицы A
```

## Анализ эффективности (по памяти и времени):

Размерность матрицы	Заполнение, % ненулевых эл-тов	Время, нс		Память, байт	
		Простая	Разреженная	Простая	Разреженная
10	1	8	20	440	136
	5	10	37		168
	20	11	53		288
	50	15	116		528
	100	14	137		928
100	1	309	109	40400	2008
	5	181	810		5208
	20	283	1592		17208
	50	429	2031		41208
	100	538	3035		81208
500	1	2704	151	1002000	26008
	5	3145	3056		106008
	20	4025	9578		406008
	50	6362	18573		1006008
	100	8364	26385		2006008
1000	1	11783	3403	4004004	92008
	5	11375	9463		412008
	20	12643	34678		1612008
	50	13043	67436		4012008
	100	13954	85276		8012028

## Вывод:

Использование разреженной матрицы выгодно при малом проценте заполнения и большой размерности матрицы (от 100). Мы получаем выигрыш по памяти при заполнении матрицы примерно до 20-30%, а выигрыш во времени примерно при 5-7%.

## Контрольные вопросы:

### 1) Что такое разреженная матрица, какие схемы хранения таких матриц вы знаете?

Разреженная матрица – матрица, большая часть которой заполнена нулями. Можно хранить индекс  $(i,j)$  ненулевого элемента и его значение .  
Можно хранить значения ненулевых элементов, их индексы строки/столбца, номера элементов, с которых начинается очередная строка.

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

Под обычную матрицу выделяется  $n*m*\text{sizeof}(\text{element})$  байт памяти.  
Требуемая память под хранение разреженной матрицы зависит от выбранного типа хранения, количества ненулевых элементов.

### 3) Каков принцип обработки разреженной матрицы?

Сократить время обработки позволяет тот момент, что обрабатываются только ненулевые элементы.

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

Эффективнее применять стандартные алгоритмы обработки матриц при достижении определенного уровня заполненности матрицы.