



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 рубежному контролю №1  
по курсу «Анализ Алгоритмов»  
на тему: «Разреженные матрицы»  
дополнение к лабораторной работе №5

Студент группы ИУ7-56Б

\_\_\_\_\_  
(Подпись, дата)

Мансуров В. М.  
(Фамилия И.О.)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Волкова Л. Л.  
(Фамилия И.О.)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Строганов Ю. В..  
(Фамилия И.О.)

Москва — 2023 г.

# Содержание

<b>Введение</b>	<b>3</b>
<b>1 Аналитическая часть</b>	<b>4</b>
1.1 Разреженный строчный формат матрицы . . . . .	4
<b>2 Конструкторская часть</b>	<b>6</b>
2.1 Требования к программному обеспечению . . . . .	6
2.2 Разработка алгоритмов . . . . .	6
<b>3 Технологическая часть</b>	<b>9</b>
3.1 Средства реализации . . . . .	9
3.2 Реализация алгоритмов . . . . .	9
<b>4 Исследовательская часть</b>	<b>12</b>
4.1 Тестирование . . . . .	12
<b>Заключение</b>	<b>14</b>
<b>Список использованных источников</b>	<b>15</b>

# Введение

Во многих областях человеческой деятельности информацию часто представляют в форме матриц. Матрица — это регулярный числовой массив. В специальной литературе имеется несколько определений разреженной матрицы. Суть их состоит в том, что матрица разрежена, если в ней больше нулевых элементов, чем ненулевых. Итак, сформулировались три основных идеи, которые направляли развитие большей части технологии разреженных матриц [1]:

- хранить только ненулевые элементы;
- оперировать только с ненулевыми элементами;
- сохранять разреженность.

Алгоритм, хранящий и обрабатывающий меньшее число нулей, более сложен, труднее программируется и целесообразен только для достаточно больших матриц.

Целью данной рубежного контроля является описание пошагового поиска строчной координаты элемента матрицы.

Для поставленной цели необходимо выполнить следующие задачи.

- 1) Описать понятие разреженной матрицы.
- 2) Описать алгоритм поиска строчной координаты элемента матрицы.
- 3) Реализовать программу, реализующую пошаговую работу алгоритма с выводом информации.

# 1 Аналитическая часть

В данном разделе рассмотрена информация, касающаяся основ конвейерной обработки данных.

## 1.1 Разреженный строчный формат матрицы

Во многих областях человеческой деятельности информацию часто представляют в форме матриц. Матрица — это регулярный числовой массив. Разреженная матрица — матрица, в которой большинство элементов равны нулю.

Разреженный строчный формат (сокр. РСФ) [1] — это одна из наиболее широко используемых схем хранения разреженных матриц. Эта схема предъявляет минимальные требования к памяти и в то же время оказывается очень удобной для нескольких важных операций над разреженными матрицами: сложения, умножения, перестановок строк и столбцов, транспонирования, решения линейных систем с разреженными матрицами коэффициентов как прямыми, так и итерационными методами и т. д. Значения ненулевых элементов матрицы и соответствующие столбцовые индексы хранятся в этой схеме по строкам в двух массивах; назовем их соответственно  $AN$  и  $JA$ . Используется также массив указателей (скажем,  $AI$ , еще обозначаемый как  $NR$ ), отмечающих позиции массивов  $AN$  и  $JA$ , с которых начинается описание очередной строки. Дополнительная компонента в  $AI$  содержит указатель первой свободной позиции в  $JA$  и  $AN$ . Пример представления матрицы РСФ на рисунке 1.1.

$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 0 & 0 & 1. & 3. & 0 & 0 & 0 & 5. & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7. & 0 & 1. & 0 & 0 \end{bmatrix} \end{matrix}$$

$A$  представляется следующим образом:

позиция	=	1	2	3	4	5	6	
IA	=	1	4	4	6			
JA	=	3	4	8	6	8		RR (C) 0
AN	=	1.	3.	5.	7.	1.		

Рисунок 1.1 – Пример представления матрицы в РСФ

В общем случае описание  $i$ -й строки хранится в позициях с  $IA(i)$  до  $IA(i + 1) - 1$  массивов  $JA$  и  $AN$ , за исключением равенства  $IA(i + 1) = IA(i)$ , означающего, что  $i$ -я строка пуста. Если матрица имеет  $n$  строк, то  $IA$  содержит  $n + 1$  позиций.

## Вывод

В данном разделе было рассмотрено понятие разреженной матрицы.

## 2 Конструкторская часть

В данном разделе будут представлены схемы последовательной и параллельной работы стадий конвейера

### 2.1 Требования к программному обеспечению

К программе предъявлены ряд требований:

- наличие интерфейса для ввода матрицы и индекс искомого элемента;
- вывод матрицы в виде РСФ после ввода, то есть трех массивов;
- вывод пошагового работы алгоритма поиска строчной координаты элемента матрицы.

### 2.2 Разработка алгоритмов

На рисунке 2.1 представлен алгоритма поиска строчной координаты элементы матрицы РСФ.

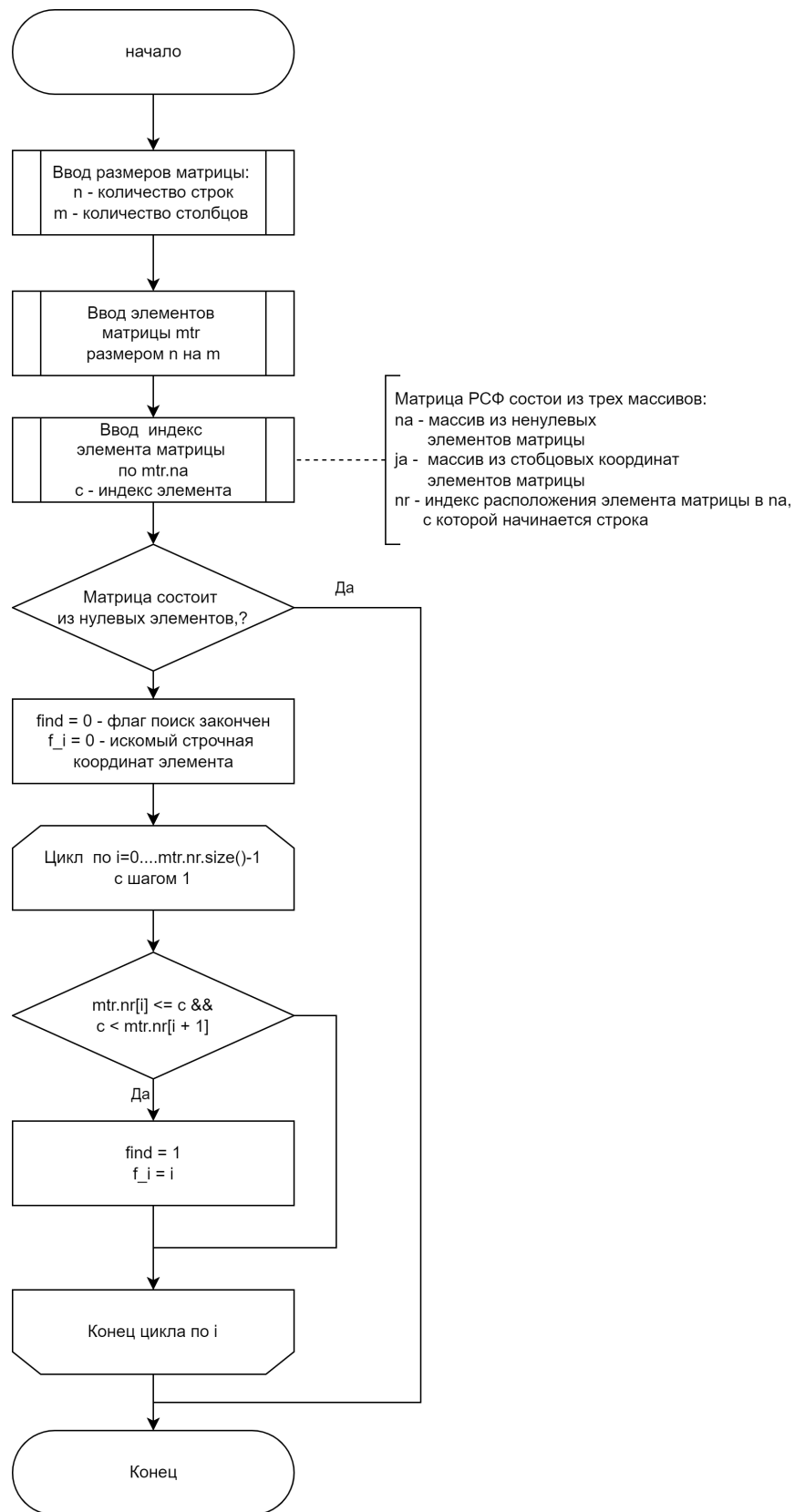


Рисунок 2.1 – Схема алгоритма поиска строчной координаты элементы матрицы РСФ

## Вывод

В данном разделе были представлены схемы последовательной и параллельной работы стадий конвейера.



## 3 Технологическая часть

В данном разделе рассмотрены средства реализации, а также представлены листинги реализаций алгоритма расчета термовой частота для всех термов из выборки документов.

### 3.1 Средства реализации

В качестве языка программирования для реализации данной лабораторной работы был выбран язык `++C` [2]. Данный выбор обусловлен тем что реализация структур данных и алгоритмов разреженной матрицы лабораторной работы №5 было выполнено на данном языке.

### 3.2 Реализация алгоритмов

На листинге 3.1 представлена структура данных *matrix<sub>c</sub>sr<sub>t</sub>*, которая задает матрицу РСФ. На листинге 3.2 представлен алгоритм поиска строчной координаты элемента матрицы с пошаговым выводом информации.

Листинг 3.1 – Реализация структуры данных матрицы РСФ

```
1 struct matrix_csr_t
2 {
3     std::vector<int> an; // элементы
4     std::vector<int> ja; // номер соответствующего столбца для эл
        емента
5     std::vector<int> nr; // номер элемента в an начала следующей
        строки — размер N + 1, N — количество строк
6
7     size_t n;
8     size_t m;
9 };
```

Листинг 3.2 – Реализация алгоритма поиска строчной координата  
элемента матрицы

```
1 int main() {
2     setbuf(stdout, nullptr);
3     matrix_csr_t mtr;
4     mtr.n = get_number("Введите количество строк n (больше 0):", 0, 1000);
5     mtr.m = get_number("Введите количество столбцов m (больше 0):", 0, 1000);
6     read_mtr_csr_t(mtr);
7     print_mtr_csr(mtr);
8     if (mtr.an.empty()) {
9         printf("Матрица полностью состоит из нулей.\n");
10        return 0;
11    }
12    int find = 0, f_i = 0;
13    int c = get_number("Введите индекс элемента (больше 0):", 1, mtr.an.size()) - 1;
14    for (int i = 0; !find && i < mtr.nr.size() - 1; i++) {
15        printf("В %d строке %d элемента(ов). NR[%d] - NR[%d] = %d", i + 1, mtr.nr[i + 1] - mtr.nr[i], i + 2, i + 1, mtr.nr[i + 1] - mtr.nr[i]);
16        if (mtr.nr[i] == mtr.nr[i + 1])
17            printf(" -> переходим к следующей строке.\n");
18        else {
19            printf(".\n");
20            if (mtr.nr[i] <= c && c < mtr.nr[i + 1]) {
21                printf("\tAN[%d] = %d находится в %d строке\n", c + 1, mtr.an[c], i + 1);
22                find = 1;
23                f_i = i;
24            } else
25                printf("\tAN[%d] = %d нет в %d строке -> переходим к следующей строке.\n", c + 1, mtr.an[c], i + 1);
26        }
27    }
28    printf("Строчной координатой элемента AN[%d] равна %d.\n", c + 1, f_i + 1);
29    return 0;
30 }
31 }
```

## Вывод

В данном разделе была приведена информация о выбранных средствах для разработки алгоритма и реализация алгоритма пошагового поиска строчной координаты .

## 4 Исследовательская часть

В данном разделе приведены примеры работы алгоритма.

### 4.1 Тестирование

```
Введите количество строк n (больше 0): 4
Введите количество столбцов m (больше 0): 8
Введите элементы матрицы:
0 0 0 0 0 0 0 0
0 0 10 0 20 0 30 0
0 0 0 0 0 0 0 0
0 0 40 50 60 0 0 0
AN = [ 10, 20, 30, 40, 50, 60 ]
JA = [ 3, 5, 7, 3, 4, 5 ]
NR = [ 1, 1, 4, 4, 7 ]
Введите индекс элемента (больше 0): 3
В 1 строке 0 элемента(ов). NR[2] - NR[1] = 0 -> переходим к следующей строке.
В 2 строке 3 элемента(ов). NR[3] - NR[2] = 3.
    AN[3] = 30 находится в 2 строке
Строчной координатой элемента AN[3] равна 2.
```

Рисунок 4.1 – Тест №1, поиск строчной координаты элемента под индексом 3, значение которого 30

```
Введите количество строк n (больше 0): 4
Введите количество столбцов m (больше 0): 8
Введите элементы матрицы:
0 0 0 0 0 0 0 0
0 0 10 0 20 0 30 0
0 0 0 0 0 0 0 0
0 0 40 50 60 0 0 0
AN = [ 10, 20, 30, 40, 50, 60 ]
JA = [ 3, 5, 7, 3, 4, 5 ]
NR = [ 1, 1, 4, 4, 7 ]
Введите индекс элемента (больше 0): 3
В 1 строке 0 элемента(ов). NR[2] - NR[1] = 0 -> переходим к следующей строке.
В 2 строке 3 элемента(ов). NR[3] - NR[2] = 3.
    AN[3] = 30 находится в 2 строке
Строчной координатой элемента AN[3] равна 2.
```

Рисунок 4.2 – Тест №2, поиск строчной координаты элемента под индексом 4, значение которого 40

```

Введите количество строк n (больше 0): 4
Введите количество столбцов m (больше 0): 8
Введите элементы матрицы:
0 0 0 0 0 0 0 0
0 0 10 0 20 0 30 0
0 0 0 0 0 0 0 0
0 0 40 50 60 0 0 0
AN = [ 10, 20, 30, 40, 50, 60 ]
JA = [ 3, 5, 7, 3, 4, 5 ]
NR = [ 1, 1, 4, 4, 7 ]
Введите индекс элемента (больше 0): 3
В 1 строке 0 элемента(ов). NR[2] - NR[1] = 0 -> переходим к следующей строке.
В 2 строке 3 элемента(ов). NR[3] - NR[2] = 3.
    AN[3] = 30 находится в 2 строке
Строчной координатой элемента AN[3] равна 2.

```

Рисунок 4.3 – Тест №3, поиск строчной координаты элемента с неверным индексом в массиве NR

# Заключение

Цель, поставленная в начале работы, была достигнута. Кроме того были достигнуты все поставленные задачи.

- 1) Описан понятие разреженной матрицы.
- 2) Описан алгоритм поиска строчной координаты элемента матрицы.
- 3) Реализовано программа, реализующую пошаговую работу алгоритма с выводом информации.

# Список использованных источников

- 1 С. Писсанецки. Технология разреженных матриц. — М.: Издательство «МИР», 1998. — С. 410.
- 2 C++ language [Электронный ресурс]. — Режим доступа:  
<https://en.cppreference.com/> (дата обращения: 28.01.2023).