



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. Описание алгоритма сложения
разреженных матриц»

Студент ИУ7-51Б
(Группа)

(Подпись, дата)

Савинова М. Г.
(Фамилия И. О.)

Преподаватель

(Подпись, дата)

Волкова Л. Л.
(Фамилия И. О.)

Преподаватель

(Подпись, дата)

Строганов Ю.В.
(Фамилия И. О.)

2024 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Аналитический раздел	4
1.1 Разреженный строчный формат матрицы	4
1.2 Сложение РСФ матриц	4
2 Конструкторская часть	6
2.1 Разработка алгоритмов	6
3 Технологический раздел	9
3.1 Средства реализации	9
3.2 Реализация алгоритмов	9
4 Исследовательский раздел	12
4.1 Демонстрация работы программы	12
ЗАКЛЮЧЕНИЕ	13
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	14

ВВЕДЕНИЕ

Целью данного рубежного контроля является описание алгоритма сложения разреженных матриц.

Для достижения поставленной цели необходимо решить следующие задачи:

- 1) описать понятие разреженной матрицы;
- 2) описать алгоритм сложения разреженных матриц;
- 3) реализовать описанный алгоритм.

1 Аналитический раздел

В данном разделе приведена информация, касающаяся разреженных матриц и графовых моделей.

1.1 Разреженный строчный формат матрицы

Разреженный строчный формат (сокр. РСФ) — это одна из наиболее широко используемых схем хранения разреженных матриц [1].

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, IA , еще обозначаемый как NR), отмечающих позиции массивов AN и JA , с которых начинается описание очередной строки. Дополнительная компонента в IA содержит указатель первой свободной позиции в JA и AN .

Рассмотрим матрицу A :

$$A = \begin{pmatrix} 0 & 0 & 0 & 1. & 3. & 0 & 0 & 5. & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7. & 0 & 1. & 0 & 0 \end{pmatrix} \quad (1.1)$$

Разреженный строчный формат матрицы 1.1 представлен ниже:

позиция	=	1	2	3	4	5	6
AN	=	1.	3.	5.	7.	1.	
JA	=	3	4	8	6	8	
NR	=	1	4	4	6		

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.

1.2 Сложение РСФ матриц

Для сложения матриц в РСФ необходимо пройти через матрицу указателей IA поэлементно через две матрицы, выделяя интервалы между соседними элементами IA по AN и JA , то есть $IA(i)$ и $IA(i+1)$.

Если встретятся элементы с одинаковыми строками и столбцами, то производится сложение. Затем происходит повторные проходы по двум матрицам по отдельности для того чтобы добавить в результирующую матрицу остаточные значения [1].

Вывод

В данном разделе было рассмотрено понятие разреженной матрицы, а также были описаны графовые модели и алгоритм сложения матриц.

2 Конструкторская часть

В данном разделе будет представлена схема алгоритма сложения разреженных матриц.

2.1 Разработка алгоритмов

На рисунке 2.1 – 2.3 представлен алгоритм сложения РСФ-матриц.

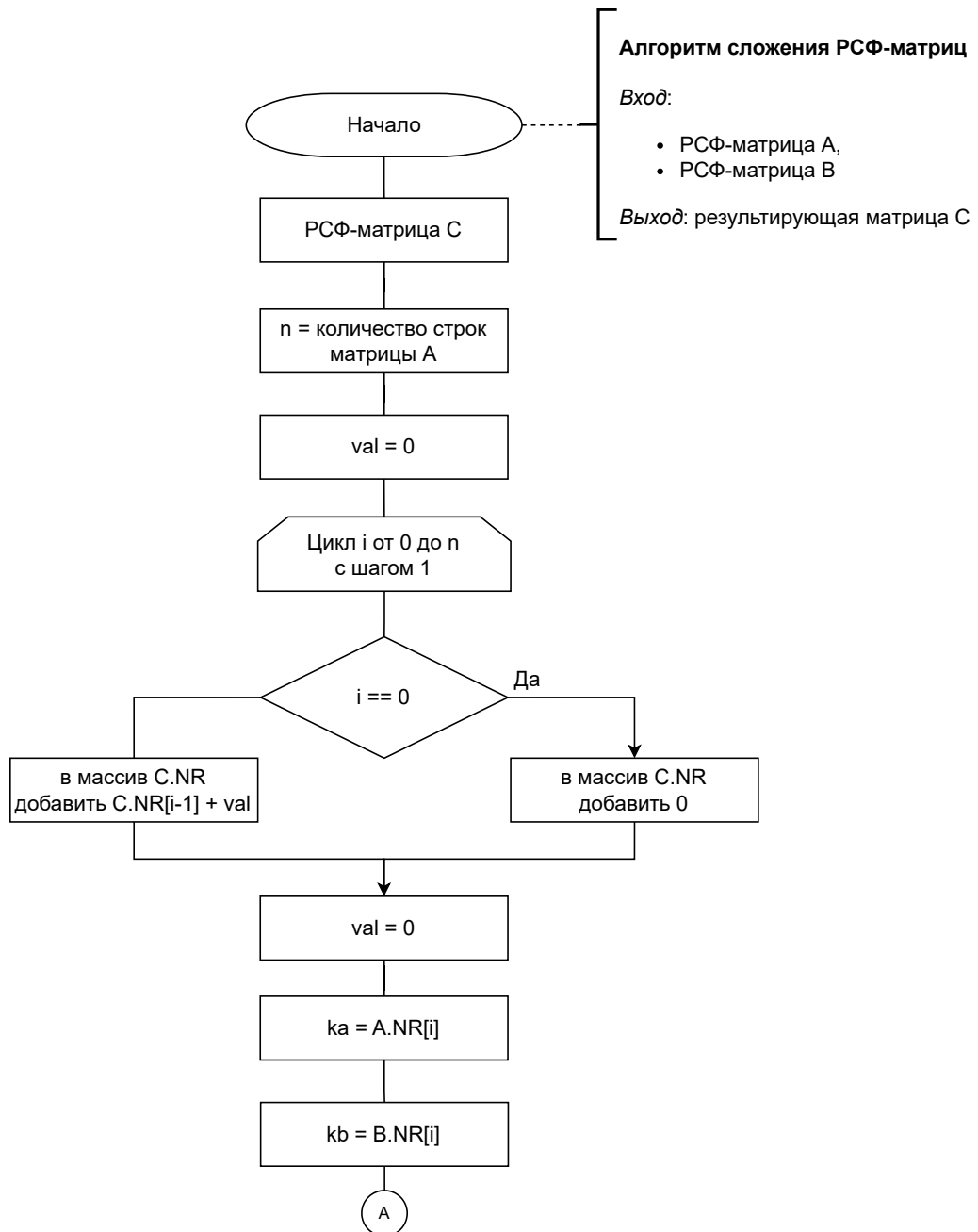


Рисунок 2.1 – Схема алгоритма сложения РСФ-матриц (часть 1)

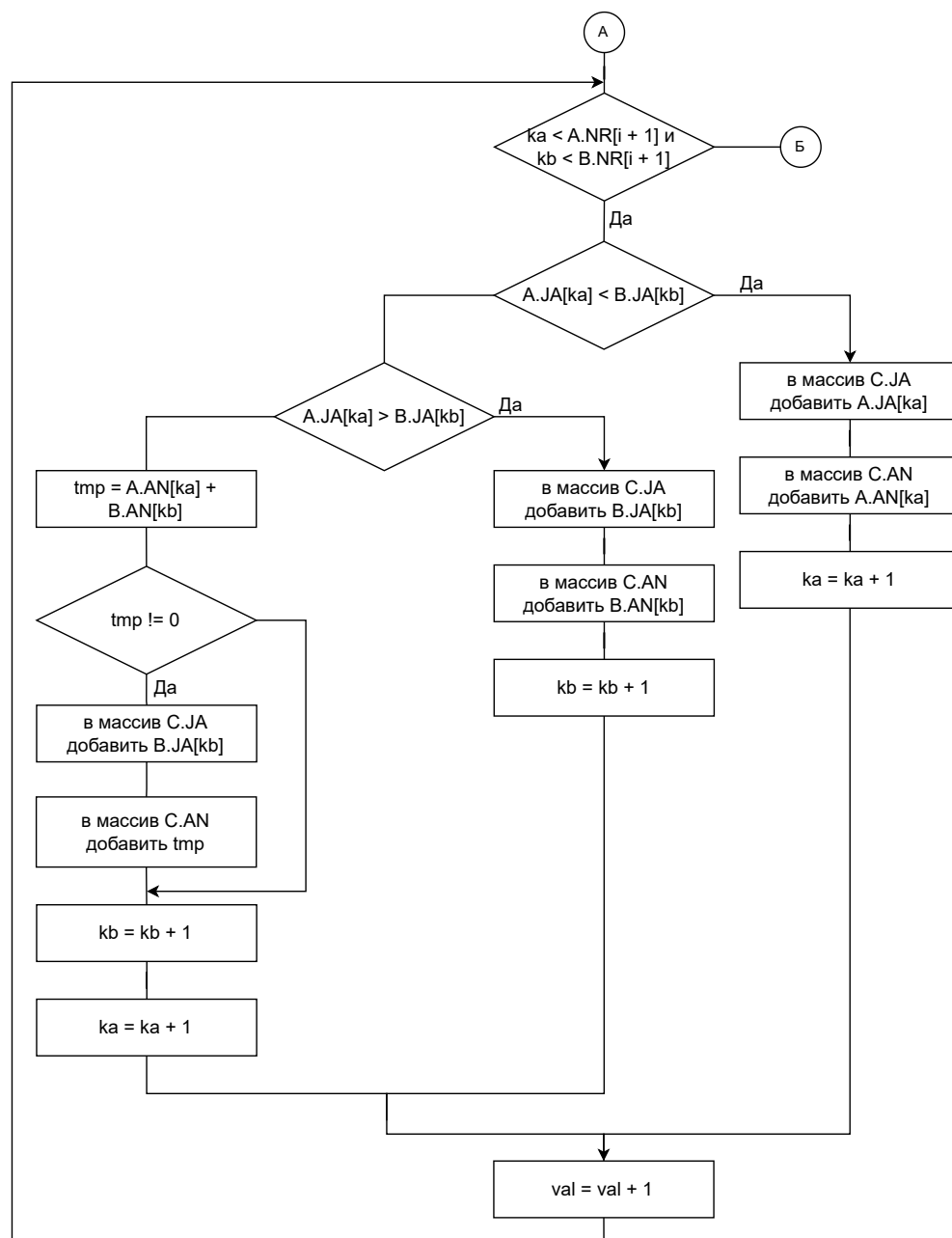


Рисунок 2.2 – Схема алгоритма сложения РСФ-матриц (часть 2)

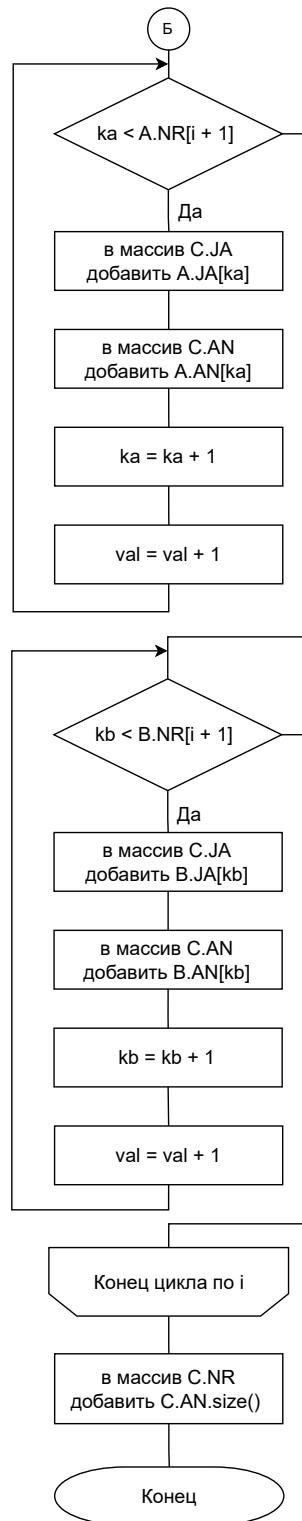


Рисунок 2.3 – Схема алгоритма сложения РСФ-матриц (часть 3)

Вывод

В данном разделе была представлена схема алгоритма сложения разреженных матриц.

3 Технологический раздел

В данном разделе будут описаны средства реализации программного обеспечения.

3.1 Средства реализации

В качестве языка программирования, используемого при написании данной лабораторной работы, был выбран C++, так как в нем имеется контейнер `std::vector`, представляющий собой динамический массив данных произвольного типа [2].

3.2 Реализация алгоритмов

На листинге 3.1 представлена реализация разрабатываемого алгоритма.

Листинг 3.1 – Реализация алгоритма сложения РСФ-матриц

```

1  MatrixCSR MatrixCSR::operator+(const MatrixCSR &mtr) {
2
3      MatrixCSR c;
4
5      c.n = n;
6      c.m = m;
7
8      int val = 0;
9
10     for (int i = 0; i < NR.size() - 1; i++) {
11
12         if (i == 0)
13             c.NR.push_back(0);
14         else
15             c.NR.push_back(c.NR[i - 1] + val);
16
17         val = 0;
18
19         int ka = NR[i];
20         int kb = mtr.NR[i];
21
22         for (; ka < NR[i + 1] && kb < mtr.NR[i + 1];) {
23
24             if (JA[ka] < mtr.JA[kb]) {
25                 c.JA.push_back(JA[ka]);
26                 c.AN.push_back(AN[ka++]);
27             } else if (JA[ka] > JA[kb]) {
28                 c.JA.push_back(mtr.JA[kb]);
29                 c.AN.push_back(mtr.AN[kb++]);
30             } else {
31                 auto tmp = AN[ka] + mtr.AN[kb];
32
33                 if (tmp) {
34                     c.JA.push_back(JA[ka]);
35                     c.AN.push_back(tmp);
36                 }
37                 ka++;
38                 kb++;
39             }
40             val++;

```

```

41         }
42
43         for (; ka < NR[i + 1]; ka++) {
44             c.JA.push_back(JA[ka]);
45             c.AN.push_back(AN[ka]);
46             val++;
47         }
48
49         for (; kb < mtr.NR[i + 1]; kb++) {
50             c.JA.push_back(mtr.JA[kb]);
51             c.AN.push_back(mtr.AN[kb]);
52             val++;
53         }
54     }
55
56     c.NR.push_back(c.AN.size());
57     return c;
58 }

```

Вывод

В данном разделе была приведена информация о выбранных средствах для разработки алгоритмов.

4 Исследовательский раздел

В данном разделе будут приведены примеры работы программы.

4.1 Демонстрация работы программы

На рисунке 4.1 представлен пример результата работы программы.

A:

AN = [8, 5, 5]

JA = [3, 0, 2]

NR = [0, 1, 3, 3, 3, 3]

B:

AN = [2, 9, 9]

JA = [0, 1, 2]

NR = [0, 3, 3, 3, 3, 3]

Итоговая матрица C:

AN = [10, 9, 9, 5, 5]

JA = [3, 1, 2, 0, 2]

NR = [0, 3, 5, 5, 5, 5]

Рисунок 4.1 – Демонстрация работы программы

ЗАКЛЮЧЕНИЕ

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

- 1) описано понятие разреженной матрицы;
- 2) описан алгоритм сложения разреженных матриц;
- 3) описанный алгоритм реализован.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *С. П.* Технология разреженных матриц //. — М.: Издательство «МИР», 1998. — С. 410.
2. C++ language. — [Электронный ресурс]. — Режим доступа: <https://en.cppreference.com/w/cpp/language> (дата обращения: 21.12.2023).