|  |  |
| --- | --- |
| 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**

**Рубежный Контроль**

**Название:**

***Разреженная матрица***

**Дисциплина:  *Анализ алгоритмов***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ***ИУ7И-56Б*** |  |  | **Нгуен Ф. С.** |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | **Волокова Л. Л.** |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

*Москва, 2020*

**Оглавление**

[Введение 3](#_Toc65278587)

[1. Аналитическая часть: 4](#_Toc65278588)

[1.1. Понятие и хранение разреженной матрицы в кольцевой схеме Рейнбольдта-Местеньи 4](#_Toc65278589)

[1.2. Описание алгоритмa склейки двух разреженных матриц 4](#_Toc65278590)

[2. Конструкторская часть 6](#_Toc65278591)

[2.1. Схемы алгоритмов 6](#_Toc65278592)

[3. Технологическая часть 9](#_Toc65278593)

[3.1. Средства реализации 9](#_Toc65278594)

[3.2. Тесты 9](#_Toc65278595)

[4. Экспериментальная часть 11](#_Toc65278596)

[4.1. Сравнение работы алгоритмов 11](#_Toc65278597)

[Заключение 14](#_Toc65278598)

# Введение

**Цель лабораторной работы** - изучение алгоритмa склейки двух разреженных матриц в кольцевой схеме Рейнбольдта-Местеньи

Для того чтобы добиться этой цели, были поставлены следующие задачи:

* изучение понятия разреженной матрицы
* хранение матриц в кольцевой схеме Рейнбольдта-Местеньи
* изучение алгоритмa склейки двух разреженных матриц
* оптимизирование работы алгоритма
* сравнение эффективности алгоритмов по времени

1. **Аналитическая часть:**
   1. Понятие и хранение разреженной матрицы в кольцевой схеме Рейнбольдта-Местеньи

**Разрежённая матрица — это [матрица](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%82%D1%80%D0%B8%D1%86%D0%B0_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)" \o ") с преимущественно нулевыми элементами.**

**Ненулевые элементы хранятся в компактной форме и в произвольном порядке в одномерном массиве, скажем AN.**

**Пусть NR <<следующий ненулевой элемент той же строки>> - Массив столбцовых указателей**

**Пусть JR JC массивы содержающине указатели входа для строк и столбцов, расположеные в соответствии с порядком строк и столбцов матрицы**

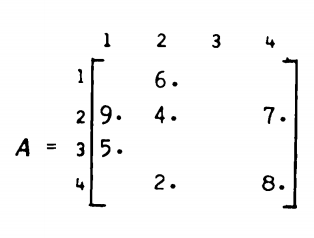
****

Рисунок 1. Разреженная матрица

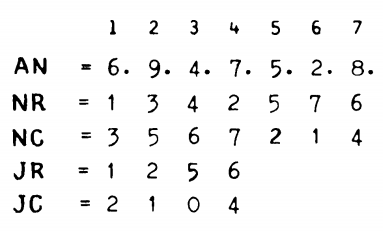
****

Рисунок 2. Хранение разреженной матрицы в кольцевой схеме Рейнбольдта-Местеньи

* 1. **Описание алгоритмa склейки двух разреженных матриц**
* AN: добавляет элементы массива 2-й матрицы AN2 к массиву AN1 2-й матрицы
* NC: увеличивает элементы в массиве NC2 второй матрицы на число, которое является количеством частей в первой матрице, затем добавьте массив NC1 массива 1
* JC: увеличивает количество элементов в массиве JC2 второй матрицы на число, которое является количеством частей в первой матрице, затем добавьте массив JC1 1-й матрицы
* Далее вычисляем 2 массива NR и JR:
  + Для каждой строки i:
    - Если 2-я строка пуста (т.е. JR2 [i] = -1), то JR [i] и NR [i] не нуждаются в изменении.
    - В противном случае, если первая строка пуста, исправить JC [i] = первый элемент второй строки
    - И наоборот, добавить вторую строку к первой строке:
      * Получим первый элемент первой строки, перейдем к концу строки,
      * Последний элемент первой строки указывает на первый элемент второй строки
      * Продолжим в конце второй строки
      * Последний элемент второй строки указывает на первый элемент первой строки
  1. *Оптимизированный алгоритм*

Для оптимизации алгоритма:

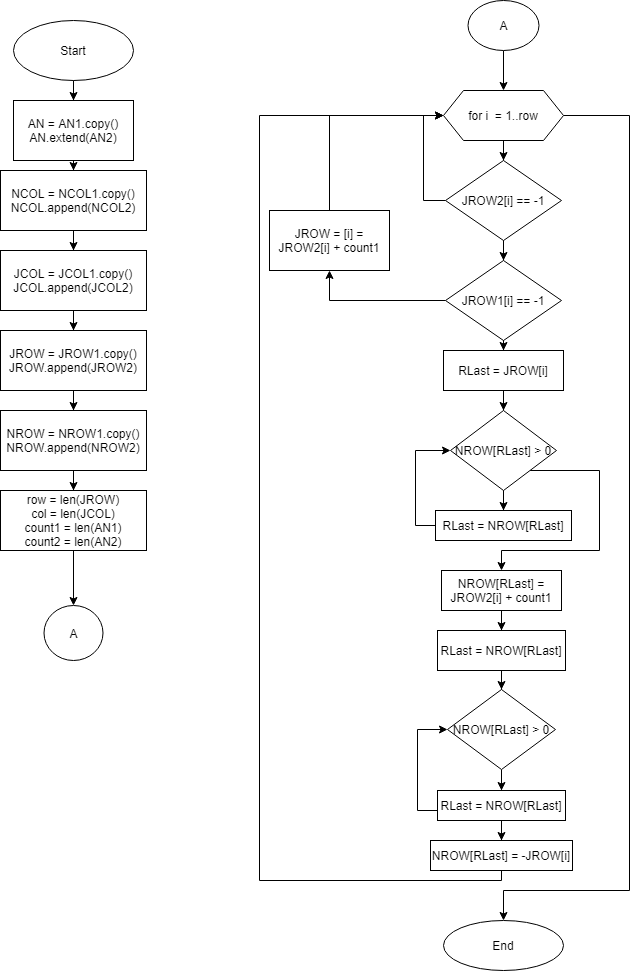
Объединить циклы с одинаковым количеством итераций (например NC и AC)

Вместо этого для каждой строки с использованием массива JC, указывающего на первый элемент каждой строки, мы используем массив lastrow, который указывает на последний элемент каждой строки, поэтому в процессе объединения двух строк нам не нужно повторять от первого до последнего мы просто используем JC [i], чтобы получить последний элемент строки i, а NC [JC [i]] дает нам первый элемент строки i. Это значительно сокращает время и объем вычислений.

1. **Конструкторская часть**
   1. **Схемы алгоритмов**

На рис. 3 приведена схема алгоритма склейки двух разреженных матриц

На рис. 4 приведена схема оптимизированного алгоритма

****

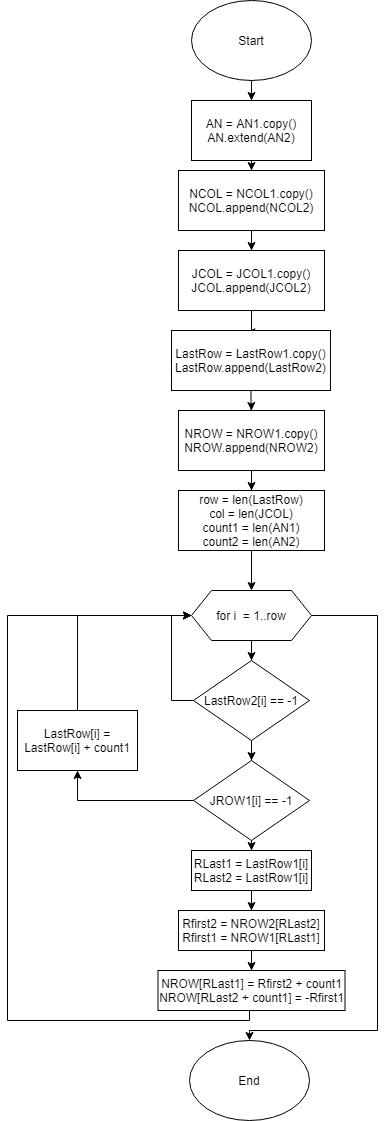
да

да

да

да

Рисунок 3. Aлгоритм склейки двух разреженных матриц

****

да

да

Рисунок 4. Оптимизированный алгоритм

1. **Технологическая часть**
   1. **Средства реализации**

Для реализации программы был использован язык Python. Для замера процессорного времени была использована функция time() из библиотеки time.

* 1. **Тесты**

Результаты тестирования показаны на рисунке 5 и 6

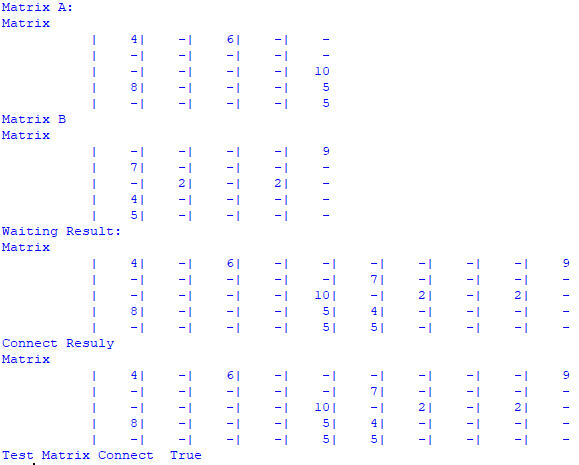
****

Figure 5. Результат программы

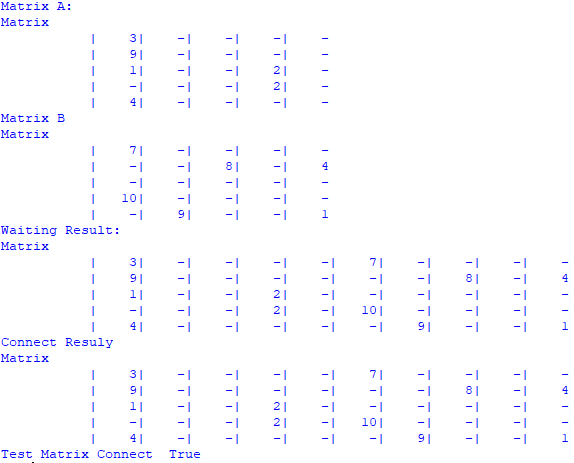
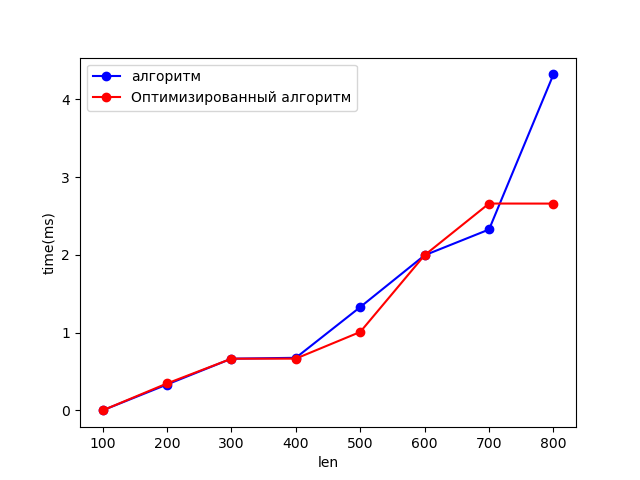
****

Figure 6. Результат программы

1. **Экспериментальная часть**
   1. **Сравнение работы алгоритмов**

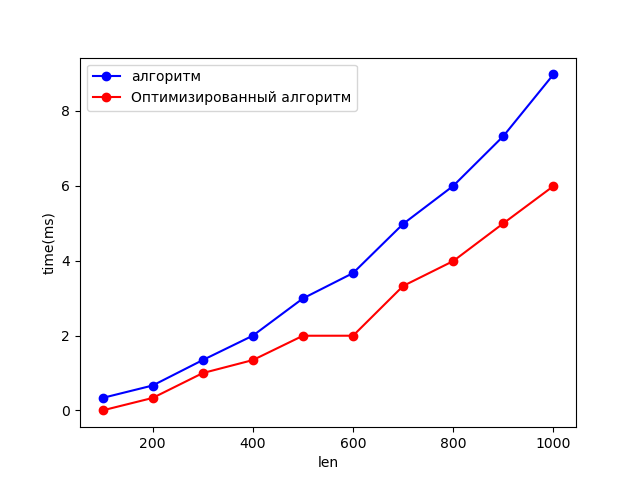
Для сравнения времени работы алгоритмов были использованы квадратные матрицы размером от 100 до 1000 с шагом 100. Результаты измерений показаны на рисунке 7-10.



Размер матрицы

время (мс)

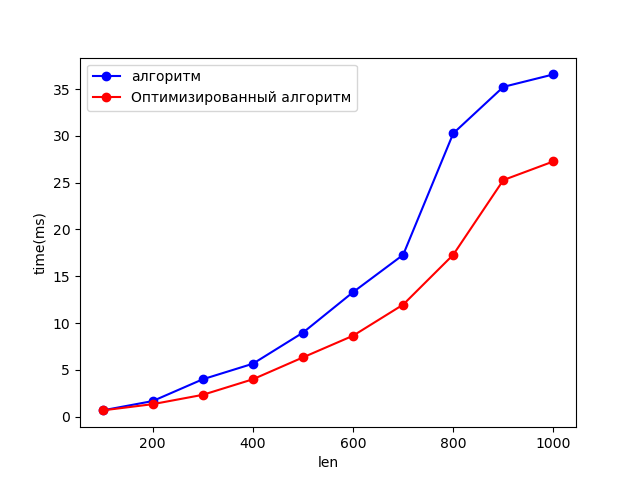
Рисунок 7. Матрица с заполнением 0,5%



время (мс)

Размер матрицы

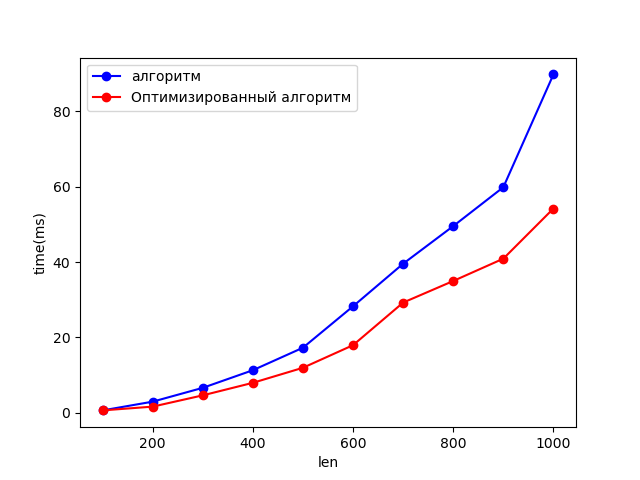
Рисунок 8. Матрица с заполнением 1%



Размер матрицы

время (мс)

Рисунок 9. Матрица с заполнением 5%



Размер матрицы

время (мс)

Рисунок 10. Матрица с заполнением 10%

* Вывод

По результатам исследования:

Когда матрица разреженная, время работы двух алгоритмов почти одинаково.

В случае высокого заполнения матрицы, тем выше разница между двумя алгоритмами.

Это выполняется путем прокрутки до конца каждого списка. Чем длиннее список (высокое заполнение), тем больше времени.

# Заключение

В рамках лабораторной работы было выполнены следующие задачи:

* изучение понятия разреженной матрицы
* хранение матриц в кольцевой схеме Рейнбольдта-Местеньи
* изучение алгоритмa склейки двух разреженных матриц
* оптимизирование работы алгоритма
* сравнение эффективности алгоритмов по времени