



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 № 2  
по курсу «Анализ алгоритмов»  
на тему: «Алгоритмы умножения матриц»

Студент ИУ7-54Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

Разин А.  
(И. О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Волкова Л. Л.  
(И. О. Фамилия)

2023 г.

# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	<b>3</b>
<b>1 Аналитическая часть</b>	<b>4</b>
1.1 Сортировка перемешиванием . . . . .	4
1.2 Поразрядная сортировка . . . . .	4
1.3 Блочная сортировка . . . . .	5
<b>2 Конструкторская часть</b>	<b>6</b>
2.0.1 Схемы алгоритмов . . . . .	6
2.0.2 Структуры данных . . . . .	6
<b>3 Исследовательская часть</b>	<b>10</b>
3.1 Технические характеристики . . . . .	10
<b>ЗАКЛЮЧЕНИЕ</b>	<b>11</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>13</b>

# ВВЕДЕНИЕ

Упорядочение данных чрезвычайно важно в программировании. Практически сортировка и поиск в той или иной мере присутствуют во всех приложениях; в частности, при обработке больших объемов данных эффективность именно этих операций определяет эффективность, а иногда и работоспособность всей системы. Под сортировкой понимается упорядочивание элементов последовательности по какому-либо признаку [1]. Можно сказать, что достаточно четкие представления об этой области нужны при решении любой задачи на ЭВМ как обязательные элементы искусства программирования [2].

Целью данной лабораторной работы является описание и исследование трудоемкости алгоритмов сортировки.

Для поставленной цели необходимо выполнить следующие задачи.

1. Описать расстояния Левенштейна и Дамерау-Левенштейна;
2. создать программное обеспечение, реализующее следующие алгоритмы сортировки
  - перемешиванием;
  - блочная;
  - поразрядная;
3. оценить трудоемкости сортировок.
4. замерить время реализации.
5. провести анализ затрат работы программы по времени, выяснить влияющие на них характеристики.

# 1 Аналитическая часть

В данной части работы будут рассмотрены алгоритмы сортировок: перемешиванием, блочная и поразрядная. Также будет определена решаемая задача.

## 1.1 Сортировка перемешиванием

Алгоритм состоит из последовательных проходов по массиву в 2 направлениях (от начала массива к его концу и наоборот). При каждом проходе происходит попарное сравнение ближайших значений, в случае если их порядок противоречит возрастающему, значения меняются местами и позиция, после которой произошло изменение порядка запоминается в переменной *left* в случае прохода от начала до конца массива и в переменной *right* иначе. Все следующие проходы по массиву происходят от позиции *left* до *right* пока данные позиции не равны. Таким образом при каждом проходе устанавливается наибольшее и наименьшее значения в индексах массива от *left* до *right* [3; 4].

## 1.2 Поразрядная сортировка

Смысл данной сортировки в том, что данные делятся сначала по разрядам и сортируются внутри каждого разряда. Сам алгоритм происходит в несколько этапов.

1. Алгоритм инициализирует индекс рассматриваемого разряда в числах;
2. после чего получает значение данного разряда каждого из чисел с помощью остатка деления на основание системы счисления;
3. затем полученные цифры сортируются;
4. элементы расставляются в соответствии со своими цифрами.

Данный алгоритм повторяется пока индекс рассматриваемого разряда не будет больше числа всех разрядов в числе [2; 4].

### 1.3 Блочная сортировка

Идея заключается в разбиении входных данных на «блоки» одинакового размера, после чего данные в блоках сортируются и результаты сортировок объединяются. Отсортированная последовательность получается путём последовательного перечисления элементов каждого блока. Для деления данных на блоки, алгоритм предполагает, что значения распределены равномерно, и распределяет элементы по блокам равномерно. Например, предположим, что данные имеют значения в диапазоне от 1 до 100 и алгоритм использует 10 блоков. Алгоритм помещает элементы со значениями 1-10 в первый блок, со значениями 11-20 — во второй, и т.д. Если элементы распределены равномерно, в каждый блок попадает примерно одинаковое число элементов. Если в списке  $N$  элементов, и алгоритм использует  $N$  блоков, в каждый блок попадает всего один или два элемента, поэтому возможно отсортировать элементы за конечное число шагов [4].

**Вывод** В данной части были рассмотрены идеи поразрядной, блочной сортировки и сортировки перемешиванием.

## 2 Конструкторская часть

В данной части работы будут рассмотрены схемы алгоритмов сортировок, а также приведен расчет их трудоемкости.

### 2.0.1 Схемы алгоритмов

### 2.0.2 Структуры данных

Для реализации выбранных алгоритмов были использованы следующие структуры данных:

1. Матрица — массив векторов типа `int`;
2. Строка — массив типа `wchar`;
3. Длина строки — целое значение типа `size_t`.

### Вывод

В данной части работы были описаны и разработаны алгоритмы поиска расстояний Левенштейна и Дамерау-Левенштейна.

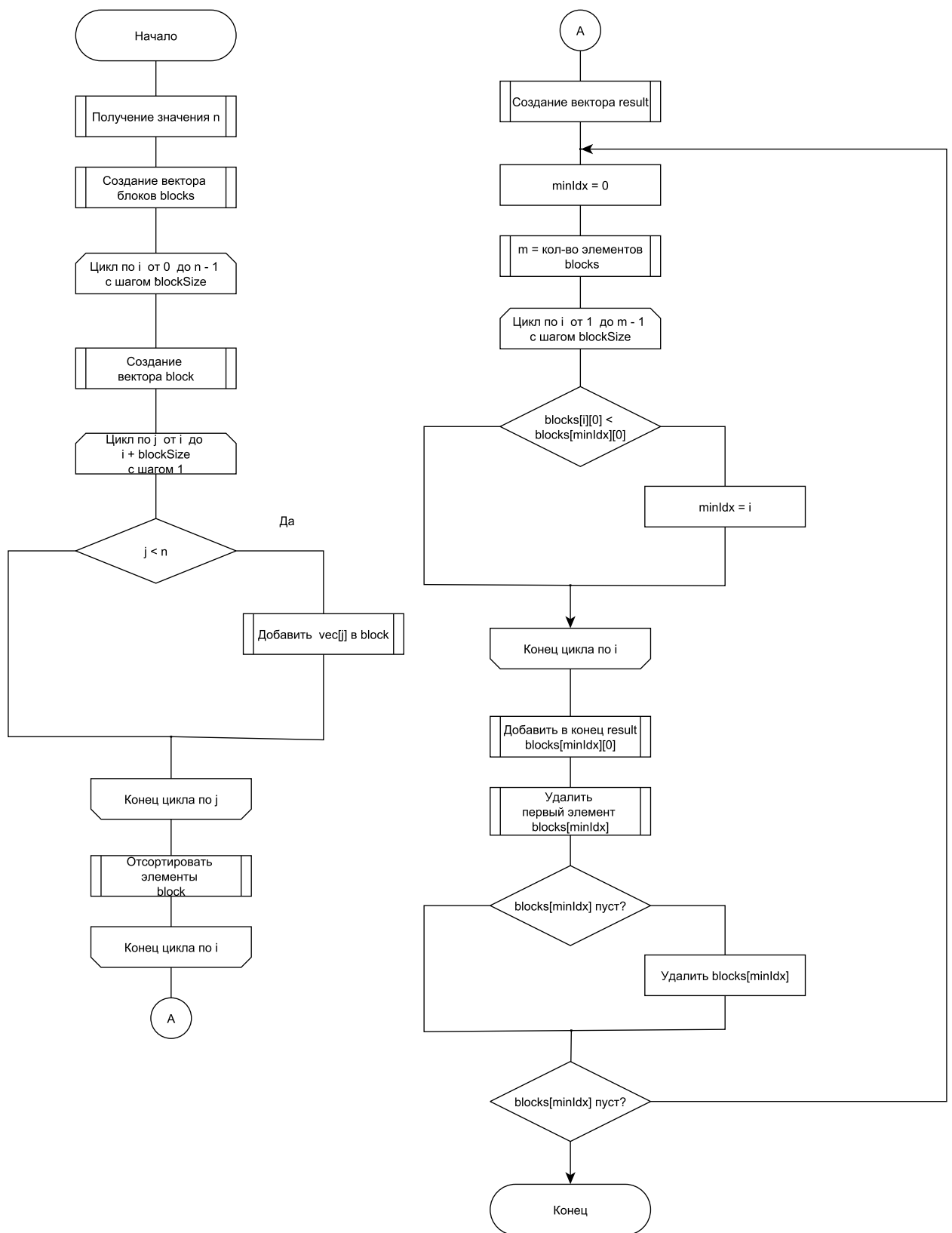


Рисунок 2.1 – Схема алгоритма сортировки перемешиванием

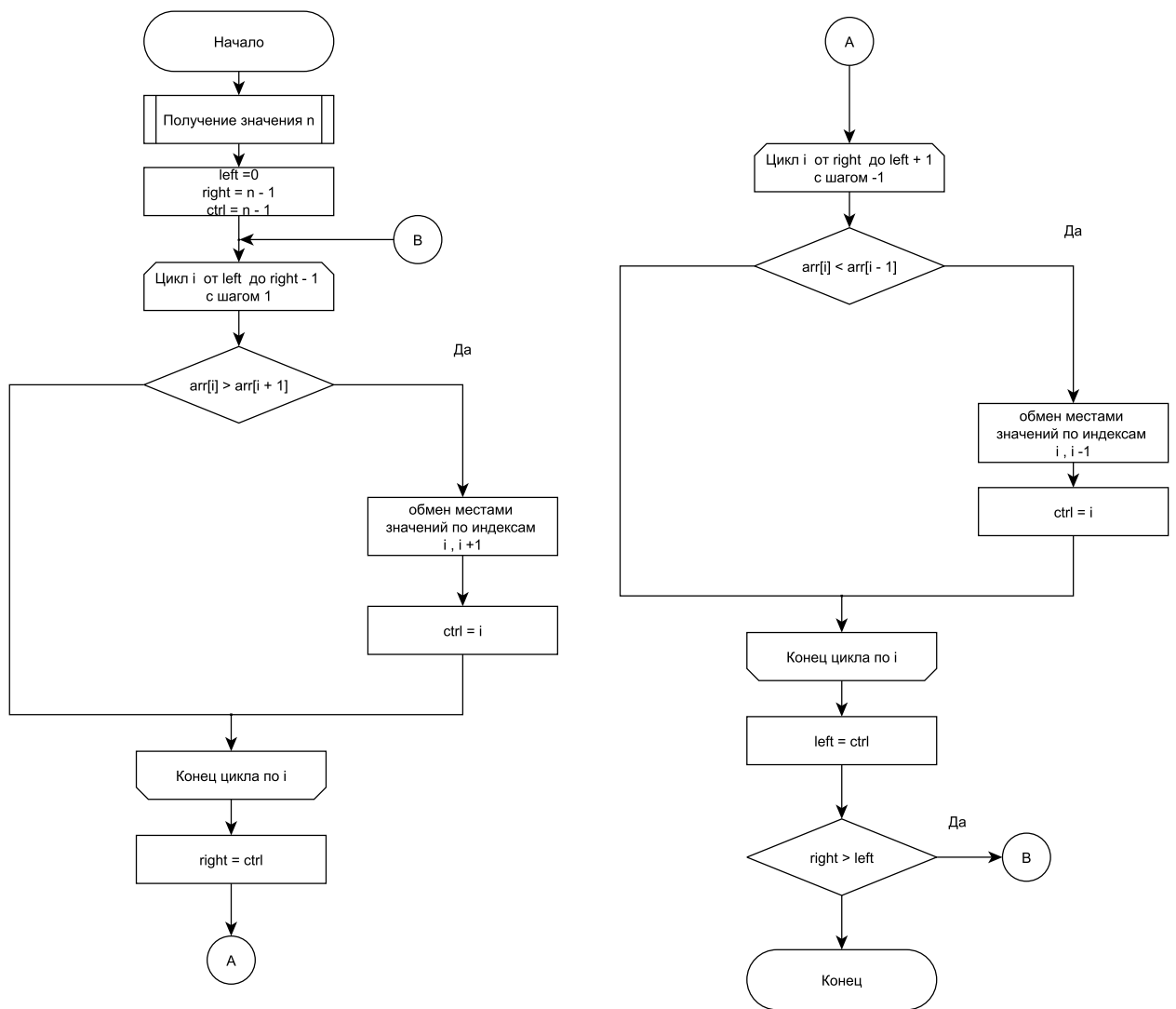


Рисунок 2.2 – Схема алгоритма блочной сортировки



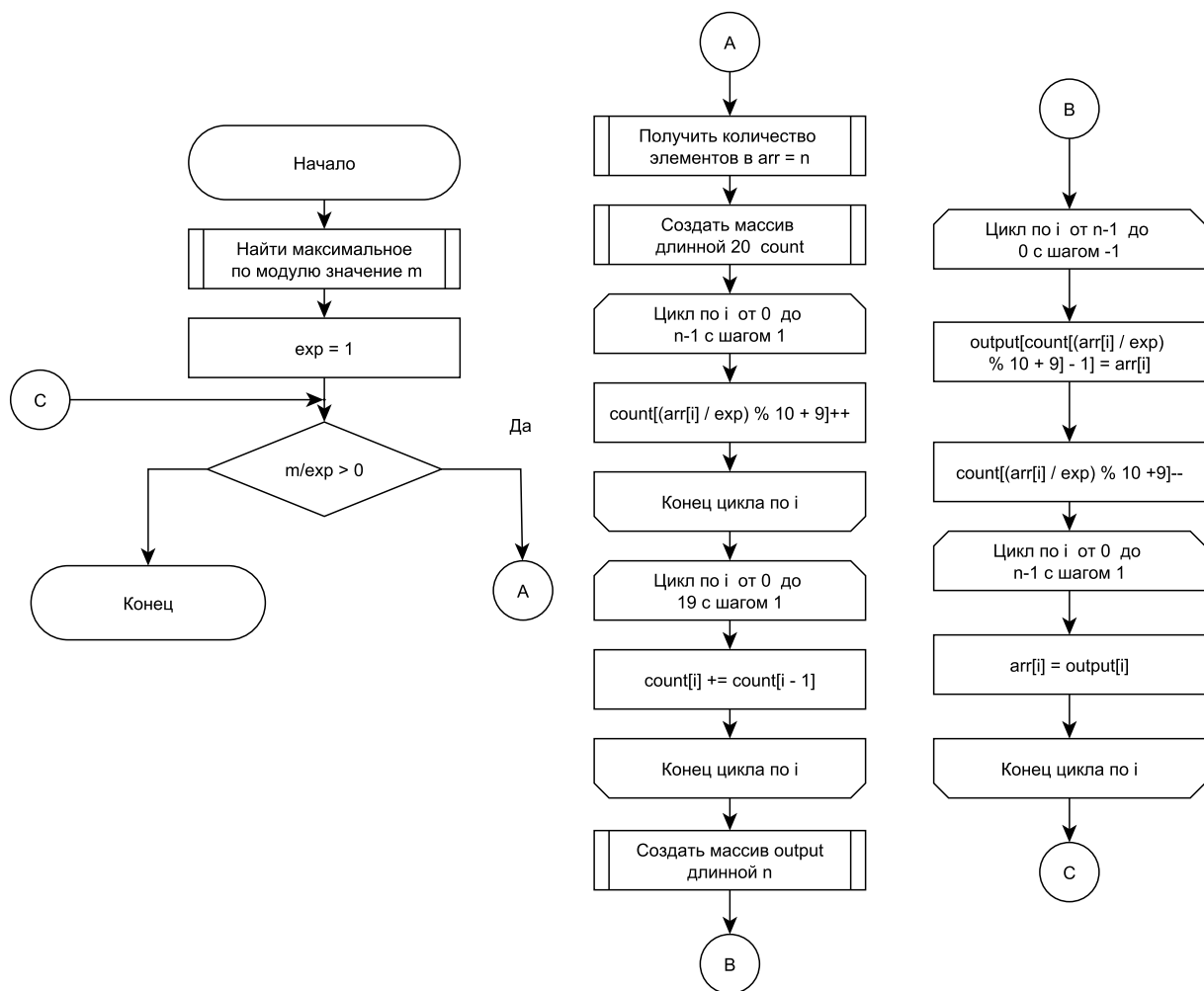


Рисунок 2.3 – Схема алгоритма поразрядной сортировки

### **3 Исследовательская часть**

#### **3.1 Технические характеристики**

## ЗАКЛЮЧЕНИЕ

В результате исследования было определено, что время алгоритмов нахождения расстояний Левенштейна и Дамерау-Левенштейна растет с ускорением при увеличении длин строк. Лучшие показатели по времени дает реализация алгоритма нахождения расстояния Дамерау-Левенштейна с использованием матрицы. При анализе таблицы ??, было установлено, что данная реализация требует в 294.662 раз меньше времени для получения расстояния, чем реализация с использованием рекурсии без мемоизации и в 1.4 раз меньше времени при использовании рекурсии с мемоизацией, при длине слов в 12 символов. При этом требования к памяти в итеративной реализации с использованием матрицы растут пропорционально произведению длин введенных строк. Из таблицы ??, можно сделать выводы, что при длине строк в 100 символов, реализации с использованием матрицы необходимо в 1.7 раз больше памяти чем реализации с помощью рекурсии без мемоизации, реализации с помощью рекурсии с мемоизацией необходимо в 1.7 больше памяти чем реализации с помощью матрицы.

Цели данной лабораторной работы были достигнуты, а именно описание и исследование особенностей задач динамического программирования на алгоритмах Левенштейна и Дамерау-Левенштейна.

Для достижения поставленной целей были выполнены следующие задачи.

- 1) Описаны алгоритмы поиска расстояния Левенштейна и Дамерау-Левенштейна.
- 2) Создано программное обеспечение, реализующее следующие алгоритмы.
  - нерекурсивный метод поиска расстояния Левенштейна;
  - нерекурсивный метод поиска расстояния Дамерау-Левенштейна;
  - рекурсивный метод поиска расстояния Дамерау-Левенштейна;
  - рекурсивный с кешированием метод поиска расстояния Дамерау-Левенштейна.
- 3) Выбраны инструменты для замера процессорного времени выполнения реализаций алгоритмов.

- 4) Проведены анализ затрат работы программы по времени и по памяти, выяснены влияющие на них характеристики.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Шагбазян., Д. В. Алгоритмы сортировки. Анализ, реализация, применение: учебное пособие / Д.В. Шагбазян, А.А. Штанюк, Е.В. Малкина. — Нижний Новгород: Нижегородский госуниверситет, 2019. — 42 с.
2. Д. К. Искусство программирования для ЭВМ. Том 3. Сортировка и поиск. // . — — М.: ООО «И.Д. Вильямс», 2014. — С. 824.
3. A comparative Study of Sorting Algorithms Comb, Cocktail and Counting Sorting [Электронный ресурс]. — — Режим доступа: [https://www.researchgate.net/profile/Ashraf-Maghari/publication/314753240\\_A\\_comparative\\_Study\\_of\\_Sorting\\_Algorithms\\_Comb\\_Cocktail\\_and\\_Counting\\_Sorting/links/58c57219aca272e36dda981b/A-comparative-Study-of-Sorting-Algorithms-Comb-Cocktail-and-Counting-Sorting.pdf](https://www.researchgate.net/profile/Ashraf-Maghari/publication/314753240_A_comparative_Study_of_Sorting_Algorithms_Comb_Cocktail_and_Counting_Sorting/links/58c57219aca272e36dda981b/A-comparative-Study-of-Sorting-Algorithms-Comb-Cocktail-and-Counting-Sorting.pdf) (дата обращения: 09.11.2023).
4. Тема 3. Компьютерный анализ данных. Лекция 10. Методы и алгоритмы обработки и анализа данных [Электронный ресурс]. — Режим доступа: [http://imamod.ru/~polyakov/arc/stud/mmca/lecture\\_10.pdf](http://imamod.ru/~polyakov/arc/stud/mmca/lecture_10.pdf) (дата обращения: 09.11.2023).