



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»

Отчёт по лабораторной работе № 3

Название: Алгоритмы сортировки

Дисциплина: Анализ алгоритмов

Студент ИУ7-55Б
(Группа)

(Подпись, дата)

Д.О. Склифасовский
(И.О. Фамилия)

Преподаватель

(Подпись, дата)

Л.Л. Волкова
(И.О. Фамилия)

Москва, 2020

Оглавление

Введение	2
1 Аналитическая часть	3
1.1 Сортировка пузырьком	3
1.2 Сортировка шейкером	3
1.3 Сортировка вставками	4
1.4 Вывод	4
2 Конструкторская часть	5
2.1 Разработка алгоритмов	5
2.2 Модель трудоемкости	8
2.3 Оценка трудоемкости алгоритмов сортировки	9
2.3.1 Сортировка пузырьком	9

Введение

Цель работы: изучение алгоритмов сортировки массивов. В данной лабораторной работе рассматриваются 3 алгоритма:

- 1) сортировка пузырьком;
- 2) сортировка шейкером;
- 3) сортировка вставками.

Также требуется изучить расчет сложности алгоритмов. В ходе лабораторной работы необходимо:

- 1) изучить алгоритмы сортировки;
- 2) дать теоритическую оценку сортировок пузырьком, шейкером и вставками;
- 3) реализовать три алгоритма сортировки на одном из языков программирования;
- 4) сравнить алгоритмы сортировки.

1 | Аналитическая часть

В данном разделе представлено описание алгоритмов сортировки массивов.

1.1 Сортировка пузырьком

Сортировка пузырьком — один из самых известных алгоритмов сортировки. Здесь нужно последовательно сравнивать значения соседних элементов и менять числа местами, если предыдущее оказывается больше последующего. Таким образом элементы с большими значениями оказываются в конце списка, а с меньшими остаются в начале.

Этот алгоритм считается учебным и почти не применяется на практике из-за низкой эффективности: он медленно работает на тестах, в которых маленькие элементы (их называют «черепашками») стоят в конце массива. Однако на нём основаны многие другие методы, например, шейкерная сортировка и сортировка расчёской.

1.2 Сортировка шейкером

Шейкерная сортировка отличается от пузырьковой тем, что она двунаправленная: алгоритм перемещается не строго слева направо, а сначала слева направо, затем справа налево.

1.3 Сортировка вставками

При сортировке вставками массив постепенно перебирается слева направо. При этом каждый последующий элемент размещается так, чтобы он оказался между ближайшими элементами с минимальным и максимальным значением.

1.4 Вывод

Было представлено описание алгоритмов сортировки массивов. В основном все алгоритмы сортировок основаны на алгоритме сортировки пузырьком.

2 | Конструкторская часть

В данном разделе представлены съемы разработанных алгоритмов. Также оценивается трудоемкость алгоритмов.

2.1 Разработка алгоритмов

На рисунке 1 изображена схема алгоритма сортировки пузырьком.

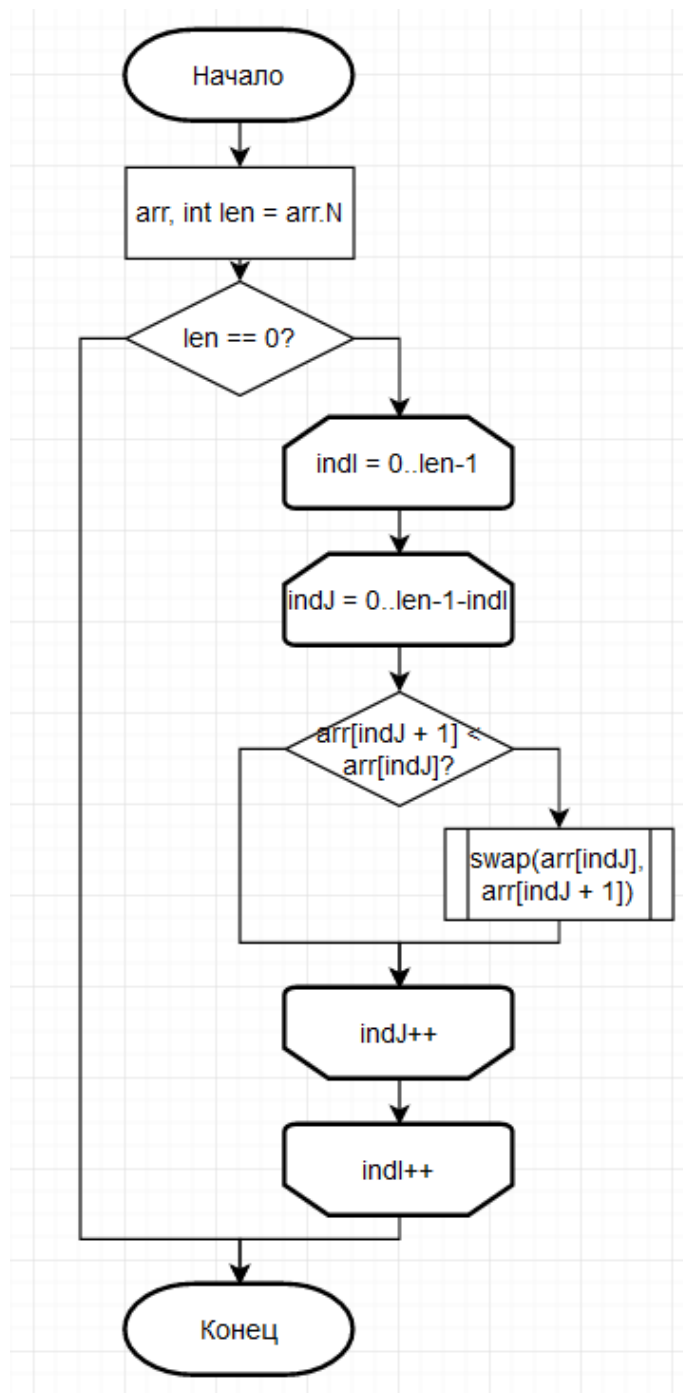


Рисунок 1. Схема алгоритма сортировки пузырьком

На рисунке 2 изображена схема алгоритма сортировки шейкером.

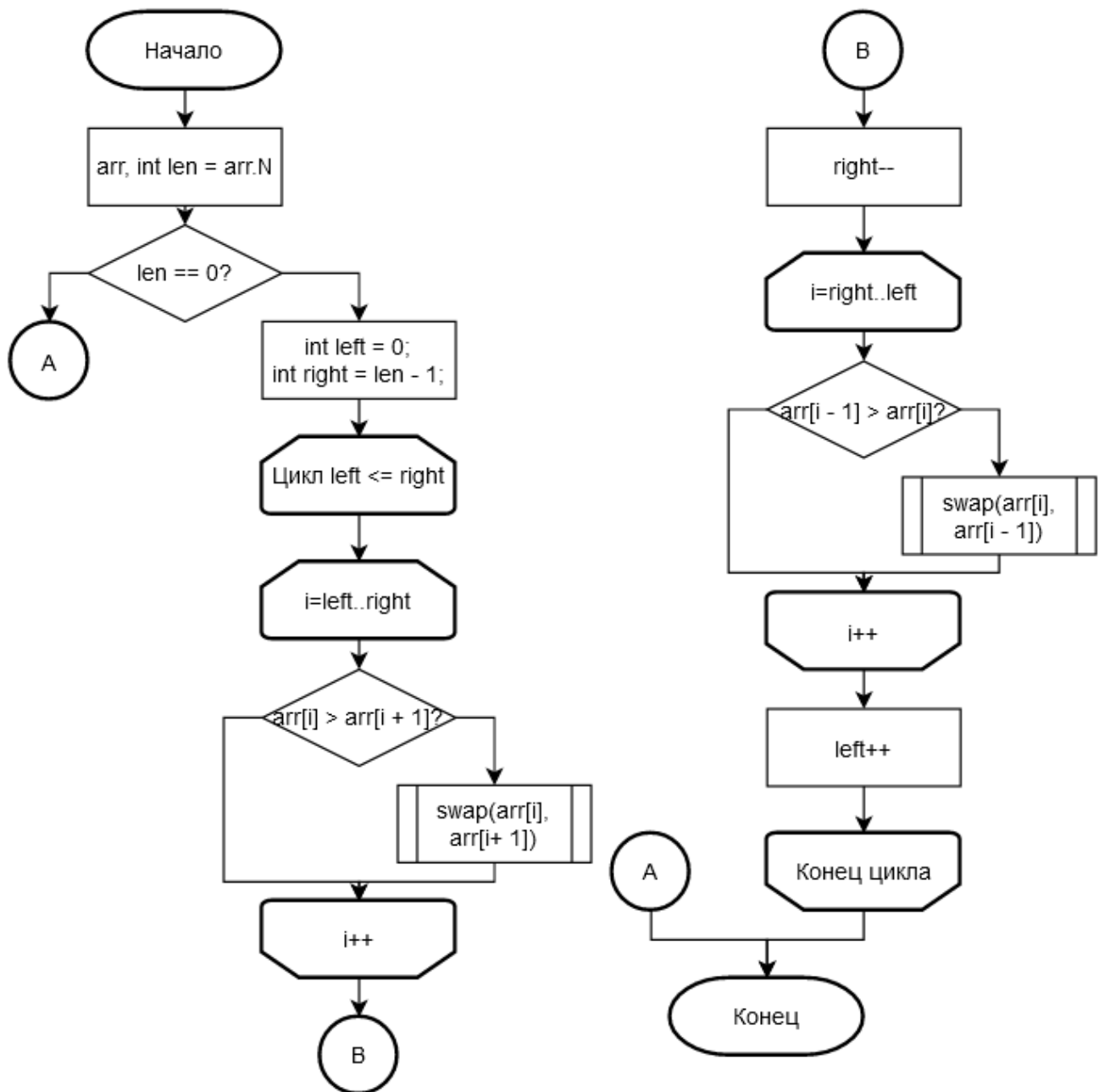


Рисунок 2. Схема алгоритма сортировки шейкером

На рисунке 3 изображена схема алгоритма сортировки вставками.

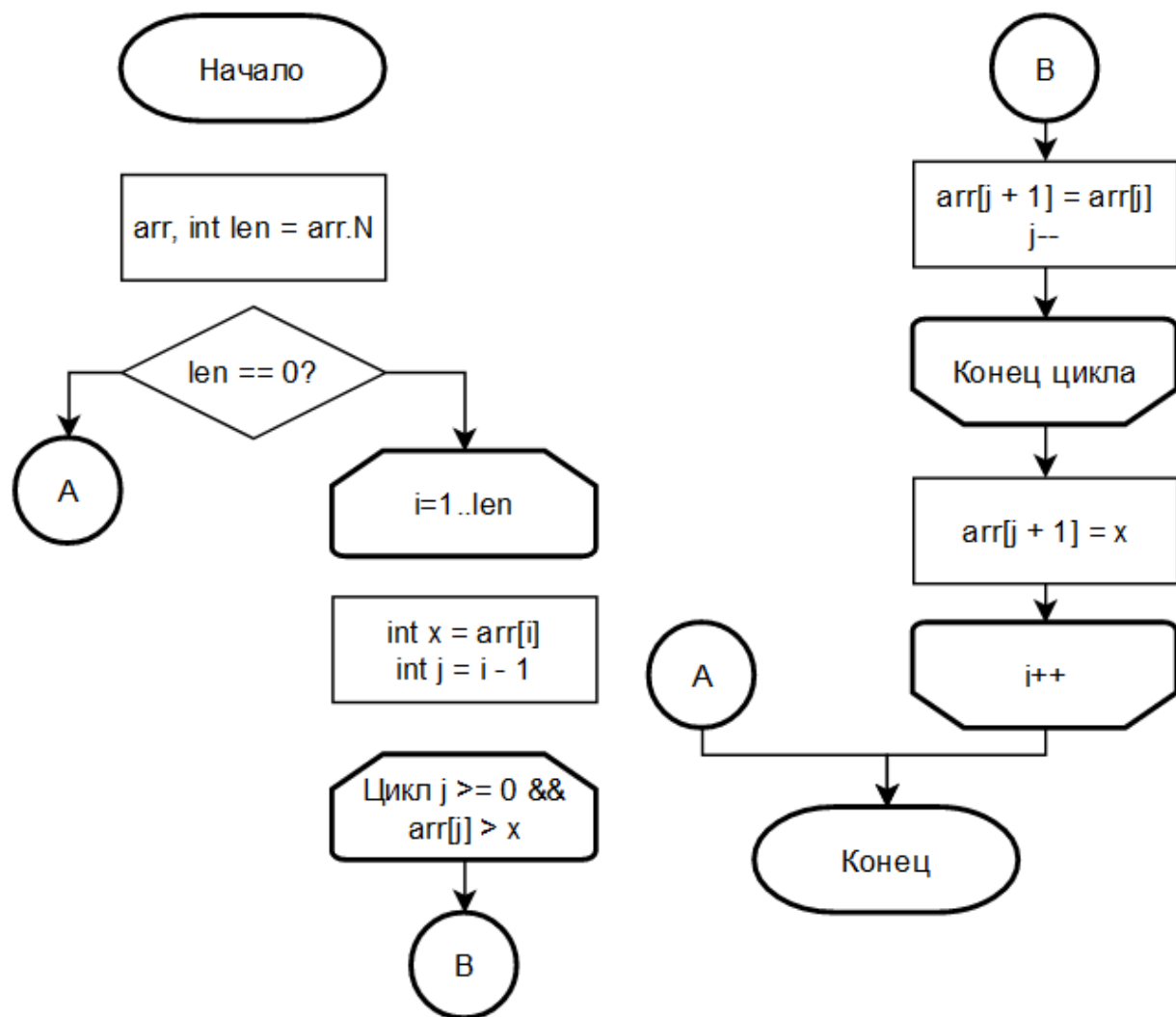


Рисунок 1. Схема алгоритма сортировки вставками

2.2 Модель трудоемкости

Модель трудоемкости для оценки алгоритмов:

1) стоимость базовых операций единица:

$=, +, *, \simeq, <, >, \geq, \leq, ==, !=, [], + =, - =, * =, / =, ++, --;$

2) стоимость цикла:

$$f_{for} = f_{init} + f_{comp} + M(f_{body} + f_{increment} + f_{comp})$$

Пример: $for(i = 0, i < M; i++) / * body * /$

Результат: $2 + M(2 + f_{body})$;

3) стоимость условного оператора

Пусть goto (переход к одной из ветвей) стоит 0, тогда

$$f_f = \begin{cases} \min(f_A, f_B), & \text{лучший случай} \\ \max(f_A, f_B), & \text{худший случай} \end{cases}$$

4) операция обращения к ячейки матрицы $[i, j]$ имеет трудоёмкость равную двум.

2.3 Оценка трудоемкости алгоритмов сортировки

Оценим трудоемкость алгоритмов.

2.3.1 Сортировка пузырьком

Лучший случай (массив отсортирован):

$$2 + 1 + 1 + 2 + (len - 1)(1 + 2 +)$$