# Самостоятельная работа 10.

# Тема. Алгоритмы сортировки

Цель:

- получить практический опыт реализации алгоритмов внутренних сортировок массива различной эффективности;
- получить навыки по выполнению анализа алгоритма сортировки с целью поучения оценки вычислительной сложности алгоритма;
- получить навыки в определение наиболее эффективного алгоритма.

## 1. Требования к выполнению практической работы

В данной практической работе требуется выполнить три задания по реализации алгоритмов сортировки с различной эффективностью. Варианты алгоритмов для реализации представлены в табл. 22.

В первом задании предлагается выполнить сортировку массива простым методом (с асимптотической сложностью $\theta(n^2)$). Определить функцию роста времени выполнения алгоритма при увеличении размера задачи для трех случаев: лучший, средний и худший. Провести эксперимент, выполняя алгоритм на компьютере, по определению временной сложности алгоритма, рассчитывая время, затраченное на выполнение алгоритма, на массивах различного размера.

Во втором задании необходимо реализовать алгоритм, который называют улучшенным, т.е. он имеет тот же метод перестановки элементов при сортировке, что и тот, который реализован и исследован в первом задании. Так же требуется, как и в задании 1, провести расчёт функции, указывающей на зависимость времени выполнения алгоритма от размера обрабатываемых данных, и эксперимент, выполняя алгоритм на компьютере для массивов разного размера, используя те же массивы значений, что и в задании 1.

В отчете по данной работе требуется сделать выводы об эффективности реализованных и исследованных в задании 1 и 2 алгоритмов.

В задании 3 требуется применить алгоритм сортировки асимптотическая сложность которого O(n logn). Асимптотическая сложность алгоритма не рассчитывается, а используется приведенная в источниках по данному методу. Провести эксперимент, аналогично заданиям 1 и 2, на тех же массивах, которые использовались в этих заданиях.

В задании 4 требуется оформить на одной координатной плоскости графика роста времени выполнения алгоритма для худшего случая всех трех алгоритмов сортировки.

## 2. Задание 1

Реализовать алгоритм и оценить зависимость времени выполнения алгоритма простой сортировки на массиве в лучшем, худшем и среднем случаях.

### 2.1. Требования к выполнению задания 1

1. Разработать и реализовать алгоритм сортировки одномерного целочисленного массива A[n] и реализовать его функцией, используя алгоритм согласно варианту, индивидуального задания - *Алгоритм задания 1*. Провести тестирование программы на исходном массиве, сформированном вводом с клавиатуры, т.е. доказать ее работоспособность.
2. Разработать функцию заполнения рабочего массива A с использованием генератора псевдослучайных чисел.
3. Провести экспериментальную оценку вычислительной сложности алгоритма, для чего выполнить контрольные прогоны программы для размеров массива n = 100, 1000, 10000, 100000 и 1000000 элементов с вычислением времени выполнения T(n) – (в миллисекундах/секундах). Полученные результаты занести в сводную таблицу, формат которой представляет табл. 21.
4. Рассчитать, по разработанному алгоритму, функцию роста - f(C+M) – зависимость количества выполняемых алгоритмом операторов (инструкций) размера задачи, от n.

Примечание. f(C+M) – это функция, указывающая зависимость количества выполняемых алгоритмом операций сравнений и перемещений от размера массива n. Рассчитывается функция по алгоритму сортировки. Т.е. получаем функцию f(n).

5. Провести эмпирическую оценку вычислительной сложности алгоритма.

В ячейку столбца $T_{эт}$ =f(C+M). помещается время, рассчитанное по формуле функции f(n), используя при расчете быстродействие процессора компьютера (тактовая частота процессора/количество выполняемых операций в секунду), на котором выполнялся алгоритм при заданном значении n (значение из столбца n соответствующей строки).

6. Провести эмпирическую оценку $T_{эп}$ фактического количества выполненных операций сравнения Сф и количества операций перемещения Мф.

Полученные результаты, представить временем, рассчитанным исходя из быстродействия процессора, вставить в сводную таблицу в столбец Тэп=Сф+Мф.

Таблица 21. Результаты эксперимента на случайно заполненном массиве

| n | T(n) время в мс/сек | $T_{эт}$=f(C+M) (сек) | $T_{эп}$=Сф+Мф (количество) |
|---|---|---|---|
| 100 | | | |
| 1000 | | | |
| 10000 | | | |
| 100000 | | | |
| 1000000 | | | |

7. Провести дополнительные прогоны программы на отсортированных массивах:

1) в строго убывающем порядке значений элементов, результаты представить в сводной таблице 2 по формату табл. 21;
2) в строго возрастающем порядке значений элементов, результаты представить в сводной таблице 3 по формату табл. 21;
3) выполнить анализ зависимости (или независимости) алгоритма сортировки от исходной упорядоченности массива и выводы представить в отчете по заданию.

8. Представить график зависимости Тэп=Сф+Мф от размера массива для лучшего, худшего, среднего случаев.
9. Провести анализ полученных результатов и указать порядок роста сложности алгоритма при увеличении размера входных данных. Сделать выводы о проделанной работе, основанные на полученных результатах.

## 3. Задание 2

Оценить зависимость времени выполнения алгоритма усовершенствованной сортировки в лучшем, худшем и среднем случаях.

### 3.1. Требования по выполнению задания

1. Разработать алгоритм усовершенствованной сортировки (задача 2), определенной в варианте, реализовать алгоритм. Сформировать таблицу Таблица 4 результатов сортировки по формату табл. 21 для массива, заполненного случайными числами.

2. Определить емкостную сложность алгоритма.
3. Определить функцию роста количества выполняемых инструкций при увеличении размера массива.
4. Провести дополнительные прогоны программы для оценки эффективности алгоритмов на сортированных массивах (наилучшем и наихудшем случаях). Сформировать таблицы 5 и 6.
5. Выполнить анализ полученных результатов по таблицам 4, 5, 6.
6. Представить график зависимости Тэп=Сф+Мф от размера массива для лучшего, худшего и среднего случаев.
7. Определить эффективный из алгоритмов (задача 1) и (задача 2) по временной и емкостной сложности.

## 4. Задание 3

Оценить зависимость времени выполнения алгоритма ускоренной сортировки в худшем, лучшем, среднем случаях.

### 4.1. Требования по выполнению задания 3

1. Разработать алгоритм ускоренной сортировки (задача 3), реализовать алгоритм.
2. Сформировать таблицу 7 результатов сортировки по формату табл. 21 для массива, заполненного случайными числами. Определить емкостную сложность алгоритма. Определить асимптотическую сложность алгоритма.

Примечание. Функцию f(C+M) не рассчитывать, а использовать уже известную асимптотическую сложность этого алгоритма.

3. Провести дополнительные прогоны программы для оценки эффективности алгоритмов в наилучшем и наихудшем случаях. Сформировать таблицы 8 и 9.
4. Представить график зависимости Сф+Мф от размера массива для лучшего, худшего и среднего случаев, не переводя во время, а используя полученное значение как количество операций.

## 5. Задание 4

Построить графики зависимости Сф+Мф (не переводя во время, а используя полученное значение как количество операций) для трех реализованных алгоритмов для наихудшего случая на одной координатной плоскости.

## 6. Варианты задач

Таблица 22. Варианты задач практической работы 10

| № | Алгоритм простой сортировки | Алгоритм усовершенствованной сортировки | Быстрый алгоритм сортировки |
|---|---|---|---|
| Задачи/ пункты | 1 | 2 | 3 |
| 1. | Простого обмена (пузырек). | Шейкерная сортировка. | Простое слияние. |
| 2. | Простого обмена (пузырек) с условием Айверсона. | Шейкерная сортировка. | Естественное слияние. |
| 3. | Простого обмена (пузырек). | Шейкерная с условием Айверсона. | Быстрая сортировка (опорный элемент последний). |
| 4. | Простой вставки. | Сортировка Шелла со смещениями Д. Кнута. Способ 1. | Простое слияние. |
| 5. | Простой вставки. | Шелла со смещениями Д. Кнута. Способ 2. | Естественное слияние. |
| 6. | Простой вставки. | Шелла со смещением Р. Седжвика. | Быстрая сортировка (опорный элемент в середине). |
| 7. | Простого выбора. | Пирамидальная сортировка. | Простое слияние. |
| 8. | Простого выбора. | Турнирная сортировка. | Естественное слияние. |
| 9. | Простой вставки. | Битонная сортировка Bitonic sort. | Рандомизированная быстрая сортировка. |
| 10. | Простой вставки. | Шелла со смещениями убывающим вдвое. | Простое слияние. |
| 11. | Простого обмена (пузырек) с условием Айверсона. | Шейкерная сортировка. | TimSort -гибридный алгоритм.. |
| 12. | Простой вставки. | Шелла со смещениями убывающим вдвое. | Многофазная сортировка слиянием |
| 13. | Простого обмена (пузырек). | Шейкерная сортировка. | Рандомизированная быстрая сортировка. |
| 14. | Простого выбора . | Пирамидальная сортировка | Сортировка подсчетом. |
| 15. | Простой вставки. | Сортировка бинарными включениями. | Сортировка Хоара. |
| 16. | Простого выбора. | Турнирная сортировка. | TimSort -гибридный алгоритм. |
| 17. | Простого обмена (пузырек) с условием Айверсона. | Шелла со смещениями убывающим вдвое. | Сортировка слиянием. |
| 18. | Простой вставки. | Пирамидальная сортировка. | Сортировка подсчетом. |

Окончание табл. 22

| № | Алгоритм простой сортировки | Алгоритм усовершенствованной сортировки | Быстрый алгоритм сортировки |
|---|---|---|---|
| Задачи/ пункты | 1 | 2 | 3 |
| 19. | Простого выбора. | Шейкерная сортировка. | Пирамидальная сортировка. |
| 20. | Простого выбора. | Сортировка бинарными включениями. | Многофазная сортировка. |
| 21. | Простого обмена (пузырек) с условием Айверсона. | Сортировка Шелла со смещениями Д. Кнута. Способ 1. | Сортировка слиянием. |
| 22. | Простой вставки. | Карманная сортировка. | Естественного слияния. |
| 23. | Простого обмена. | Сортировка подсчетом. | Быстрая сортировка с опорным последним элементом. |
| 24. | Простой вставки. | Шейкерная с условием Айверсона. | Сортировка Хоара. |
| 25. | Простого обмена. | Сортировка Шелла со смещения ми по Седжвику. | Пирамидальная сортировка. |
| 26. | Простой вставки. | Сортировка бинарными включениями. | Турнирная сортировка. |
| 27. | Простого обмена (пузырек) с условием Айверсона. | Шелла со смещением Кнута, способ 1. | Простого слияния. |
| 28. | Простого выбора. | Пирамидальная сортировка. | Естественного слияния. |
| 29. | Простой вставки. | Поразрядная сортировка. | Быстрая рандомизированная сортировка. |
| 30. | Простого обмена. | Сортировка подсчетом. | Пирамидальная. |

## 7. Структура отчета

Титул.

Оглавление.

1. Определить АТД задачи, включив:

1.1. Данные:

– массив элементов целого типа;

– размер массива.

1.2. Операции:

– создание динамического массива из n элементов;

– заполнение массива случайными числами;

– заполнение массива значениями, упорядоченными по возрастанию;

– заполнение массива значениями, упорядоченными по убыванию;

– сортировка первым методом;

– сортировка вторым методом;

– сортировка третьим методом.

2. Задание 1.

2.1. Код программы задания 1.

2.2. Вывод функции f(C+M).

Представить код в таблице, указывая для каждой строки кода количество выполняемых операций. Как в лекции.

2.3. Сводная таблица 1.

2.4. Графики по трем столбцам таблицы 1 на одной координатной плоскости.

2.5. Таблицы 2, 3.

2.6. Графики случаев: лучший, худший, средний на одной координатной плоскости.

2.7. Выводы о зависимости времени выполнения алгоритма от размера входных данных и упорядоченности массива. Асимптотическая сложности алгоритма для трех случаев.

2.8. Зависимость объема памяти от размера обрабатываемых данных. Нотация Big-O.

3. Отчет по заданию 2.

3.1. Код программы задания 2.

3.2. Таблицы 4, 5, 6.

3.3. Графики по трем столбцам таблицы 4 на одной координатной плоскости.

3.4. Графики случаев: лучший, худший, средний на одной координатной плоскости.

3.5. Выводы о зависимости времени выполнения алгоритма от размера входных данных и упорядоченности массива. Асимптотическая сложность алгоритма для трех случаев.

3.6. Зависимость объема памяти от размера обрабатываемых данных. Нотация Big-O.

4. Отчет по заданию 3.

4.1. Код программы задания 3.

4.2. Таблицы 7, 8, 9.

4.3. Графики по трем столбцам таблицы 7 на одной координатной плоскости.

4.4. Графики случаев: лучший, худший, средний на одной координатной плоскости.

4.5. Выводы о зависимости времени выполнения алгоритма от размера входных данных и упорядоченности массива. Асимптотическая сложности алгоритма для трех случаев.

4.6. Зависимость объема памяти от размера обрабатываемых данных. Нотация Big-O.

5. Отчет по заданию 4.

5.1. Представить график зависимости Сф+Мф для трех реализованных алгоритмов для наихудшего случая.

5.2. Указать наиболее эффективный алгоритм, исходя из графиков, указывающих скорость роста времени. Указать значение n0, начиная с которого скорость роста времени какого-то алгоритма возрастает.

5.3. Привести асимптотическую сложность каждого из трех алгоритмов для лучшего, худшего и среднего случая.

6. Выводы

Заполнить таблицу для исследованных алгоритмов.

| Название алгоритма | Асимптотическая сложность алгоритма | | | |
|---|---|---|---|---|
| | Наихудший случай | Наилучший случай | Средний случай | Емкостная сложность |
| | | | | |
| | | | | |
| | | | | |

## 8. Методы Д. Кнута и Р. Седжвика определения смещения для сортировки методом Шелла

Перед выполнением сортировки происходит вычисление длин промежутков, которые записываются в массив, например, d.

По методу Р. Седжвика.

Значение смещения, записываемого в элемент массива d, вычисляется по формуле:

$$k[i] = \begin{cases} 9 * 2^{i} - 9 * 2^{i/2} + 1 \text{ при i - четном} \\ 8 * 2^{\text{i}} - 6 * 2^{(i+1)/2} + 1 \text{ при i нечетном} \end{cases}$$

(1)

Остановить создание и заполнение массива d на значении d[i-1], если 3*d[i]>n (размера массива).

По методу Д. Кнута.

Определение длины промежутков методом предложенным Кнутом.

Способ 1:

$t=\log_3 n-1$

(2)

где $d_t=1$, $d[i-1]=3*d[i]+1$ т.е. 1, 4, 13, 40, 121, …..

Способ 2:

$t=\log_2 n-1$ (3)

где $d_t=1$, $d[i-1]=2*d[i]+1$ т.е. 1, 3, 7, 13, 31, …..

**9. Контрольные вопросы**

1. Количество каких операций алгоритма сортировки, подсчитывается при анализе вычислительной сложности алгоритма сортировки?
2. Какой метод отличает алгоритмы простых внутренних сортировок массива?
3. Какое действие выполняет алгоритм сортировки массива A из n элементов методом “Простого выбора” (SelectionSort), сортируя массив по возрастанию, на j-ом проходе?
4. Сколько сравнений и сколько перемещений выполнит алгоритм сортировки “Простая вставка” (InsortionSort)? сортируя массив 4 5 7 8 3 2 1 по возрастанию значений?
5. Сколько сравнений и сколько перемещений выполнит алгоритм сортировки “Простого обмена”, сортируя массив 4 5 7 8 3 2 1 по возрастанию значений и перебирая значения в направлении от последнего элемента к первому?
6. Сколько проходов по массиву не активизируют перемещения элементов (лишние) в алгоритме сортировки “Простого обмена”, если сортируется массив 4 5 7 8 3 2 1 по возрастанию значений, просматривает значения в направлении от последнего элемента к первому?
7. Как повысить эффективность алгоритма сортировки “Простого обмена”?
8. В чем идея алгоритма Айверсона?
9. Чем алгоритм сортировка методом Шелла отличается от алгоритма сортировки методом «Простой вставки»?
10. Какова вычислительная сложность алгоритмов сортировки: «Простого выбора», «Простого обмена», «Простой вставки» в лучшем случае?
11. Алгоритм «Простого слияния» или «Простой вставки» быстрее завершит сортировку массива, упорядоченного по возрастанию? Объясните ваши выводы.

12. В каком случае алгоритм быстрой сортировка имеет асимптотическую сложность $O(n^2)$?
13. В чем идея метода сортировки «Естественное слияние»?
14. Изобразите процесс сортировки по возрастанию массива 5 7 2 8 1 методом «Пирамидальная сортировка».
15. Выполните сортировку по возрастанию массива 5 5 4 1 2 4 2 1, применяя метод «Подсчета»?
16. Для каких массивов применение сортировки подсчетом не эффективно?
17. Какие дополнительные структуры хранения значений использует алгоритм «Поразрядная сортировка»?