

## ПРАКТИЧЕСКАЯ РАБОТА №2

### ЭМПИРИЧЕСКИЙ АНАЛИЗ СЛОЖНОСТИ ПРОСТЫХ АЛГОРИТМОВ СОРТИРОВКИ

**Цель.** Актуализация знаний и приобретение практических умений по эмпирическому определению вычислительной сложности алгоритмов.

**Задание 1.** Оценить эмпирически вычислительную сложность алгоритма простой сортировки на массиве, заполненном случайными числами (средний случай).

1. Составить функцию простой сортировки одномерного целочисленного массива  $A[n]$ , используя алгоритм согласно варианту индивидуального задания (столбец *Алгоритм заданий 1 и 2* в таблице 1). Провести тестирование программы на исходном массиве  $n=10$ .

2. Используя теоретический подход, определить для алгоритма:

- а. Что будет ситуациями лучшего, среднего и худшего случаев.
- б. Функции роста времени работы алгоритма от объёма входа для лучшего и худшего случаев.

**Таблица 1. Варианты индивидуальных заданий**

№	Алгоритм заданий 1 и 2	Алгоритм задания 3
1	Простой вставки ( <i>Insertion sort</i> )	Простого выбора ( <i>Selection sort</i> )
2	Простого обмена («пузырек», <i>Exchange sort</i> )	Простой вставки ( <i>Insertion sort</i> )
3	Простого выбора ( <i>Selection sort</i> )	Простого обмена («пузырек», <i>Exchange sort</i> )

3. Провести контрольные прогоны программы массивов случайных чисел при  $n = 100, 1000, 10000, 100000$  и  $1000000$  элементов с вычислением времени выполнения  $T(n)$  – (в миллисекундах/секундах). Полученные результаты свести в сводную таблицу 2.

4. Провести эмпирическую оценку вычислительной сложности алгоритма, для чего предусмотреть в программе подсчет фактического количества критических операций  $T_n$  как сумму сравнений  $C_n$  и перемещений  $M_n$ . Полученные результаты вставить в сводную таблицу 2.

**Примечание:** столбец  $T_T$  для среднего случая оставим незаполненным.

**Таблица 2. Сводная таблица результатов**

n	$T(n)$ , мс	$T_T = C + M$	$T_n = C_n + M_n$
100			
1000			
10000			
100000			
1000000			

5. Построить график функции роста  $T_n$  этого алгоритма от размера массива  $n$ .
6. Определить ёмкостную сложность алгоритма.
7. Сделать вывод об эмпирической вычислительной сложности алгоритма на основе скорости роста функции роста.

**Задание 2.** Оценить вычислительную сложность алгоритма простой сортировки в наихудшем и наилучшем случаях.

1. Провести дополнительные прогоны программы на массивах при  $n = 100, 1000, 10000, 100000$  и  $1000000$  элементов, отсортированных:
  - а. строго в убывающем порядке значений, результаты представить в сводной таблице по формату *Таблицы 2*;
  - б. строго в возрастающем порядке значений, результаты представить в сводной таблице по формату *Таблицы 2*;
2. Сделать вывод о зависимости (или независимости) алгоритма сортировки от исходной упорядоченности массива.

**Задание 3.** Сравнить эффективность алгоритмов простых сортировок

1. Выполнить разработку и программную реализацию второго алгоритма согласно индивидуальному варианту в столбце *Алгоритм задания 3* из таблицы 1.
2. Аналогично заданиям 1 и 2 сформировать таблицы с результатами эмпирического исследования второго алгоритма в среднем, лучшем и худшем случаях в соответствии с форматом Таблицы 2 (на тех же массивах, что и в заданиях 1 и 2).
3. Определить ёмкостную сложность алгоритма от  $n$ .
4. На одном сравнительном графике отобразить функции  $T_n(n)$  двух алгоритмов сортировки в худшем случае.
5. Аналогично на другом общем графике отобразить функции  $T_n(n)$  двух алгоритмов сортировки для лучшего случая.
6. Выполнить сравнительный анализ полученных результатов для двух алгоритмов.

Требования к выполнению заданий 1-3:

**1. В отчёте по заданию 1:**

- 1.1. Привести алгоритм сортировки по методу варианта *Алгоритм заданий 1 и 2* на случайно заполненном массиве. Представить результаты тестирования при  $n=10$ .
- 1.2. Описать процесс определения функции роста метода сортировки.
- 1.3. Представить сводную таблицу результатов выполнения сортировки по

указанным объемам по формату *Таблицы 2* на случайно заполненном массиве для всех указанных объемов.

- 1.4. Представить график согласно заданию, выполненный с помощью любого вспомогательного ПО. График должен: быть подписан, быть наглядным, с подписанными и размеченными координатными осями.

## **2. В отчёте по заданию 2.**

- 2.1. Представить сводную таблицу результатов при применении алгоритма к массиву, упорядоченному: по возрастанию, по убыванию.
- 2.2. Представить код программы и результат работы при  $n=10$ .
- 2.3. Сделать вывод о вычислительной сложности алгоритма.

## **3. В отчёте по заданию 3.**

1. Привести алгоритм сортировки по методу *Алгоритм задания 3*. Представить результаты тестирования при  $n=10$ .
2. Описать процесс определения функции роста метода сортировки.
3. Представить сводные таблицы результатов выполнения сортировки по указанным объемам по формату *Таблица 2*. Для среднего, худшего и лучшего случаев.
4. Представить графики в соответствии с заданиями. Каждый график должен: быть подписан, быть наглядным, с подписанными и размеченными координатными осями.
5. Сделать вывод о том, какой из алгоритмов эффективнее на основе данных эмпирического анализа.

### **Вопросы для самоподготовки**

1. Какие сортировки называют простыми?
2. Что означает понятие «внутренняя сортировка»?
3. Какие операции считаются основными при оценке сложности алгоритма сортировки?
4. Какие характеристики сложности алгоритма используются при оценке эффективности алгоритма?
5. Какая вычислительная и емкостная сложность алгоритма: простого обмена, простой вставки, простого выбора?
6. Какую роль в сортировке обменом играет условие Айверсона?
7. Определите, каким алгоритмом, рассмотренным в этом задании, сортировался исходный массив 5 6 1 2 3. Шаги выполнения сортировки:
  - 1) 1 5 6 2 3
  - 2) 1 2 5 6 3
  - 3) 1 2 3 5 6

8. Какова вычислительная теоретическая сложность алгоритма сортировки, рассмотренного в вопросе 7.

### **ЛИТЕРАТУРА:**

1. Бхаргава А. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих. – СПб: Питер, 2017. – 288 с.
2. Вирт Н. Алгоритмы + структуры данных = программы. – М.: Мир, 1985. – 406 с.
3. Кнут Д.Э. Искусство программирования, том 3. Сортировка и поиск, 2-е изд. – М.: ООО «И.Д. Вильямс», 2018. – 832 с.
4. Седжвик Р. Фундаментальные алгоритмы на C++. Анализ/Структуры данных/Сортировка/Поиск. – К.: Издательство «Диасофт», 2001. – 688 с.
5. AlgoList – алгоритмы, методы, исходники [Электронный ресурс]. URL: <http://algotlist.manual.ru/> (дата обращения 15.03.2022).
6. Алгоритмы – всё об алгоритмах / Хабр [Электронный ресурс]. URL: <https://habr.com/ru/hub/algorithms/> (дата обращения 15.03.2022).
7. НОУ ИНТУИТ | Технопарк Mail.ru Group: Алгоритмы и структуры данных [Электронный ресурс]. URL: <https://intuit.ru/studies/courses/3496/738/info> (дата обращения 15.03.2022).