

Гречишкина Е.И.

студентка

Белгородский Государственный национальный

исследовательский университет

Россия, г. Белгород

Гурьянова И.В.

старший преподаватель

Белгородский Государственный национальный

исследовательский университет

Россия, г. Белгород

АНАЛИЗ ШЕЙКЕР-СОРТИРОВКИ В МАССИВАХ

Статья посвящена анализу шейкер-сортировки и выяснению зависимости времени работы алгоритма от количества элементов и их упорядочивания в массиве.

Ключевые слова: сортировка, время, упорядочивание, шейкер-сортировка, сортировка «пузырьком».

Article is devoted to the analysis a shaker sorting and to clarification of dependence of operating time of an algorithm on quantity of elements and their order in the massiv.

Keywords: sorting, time, ordering, shaker sorting, sorting by "bubble".

Для решения многих задач удобно сначала упорядочить данные по определенному признаку, так можно ускорить поиск некоторого объекта.

Сортировка – один из важнейших аспектов обработки данных, позволяющий ускорить и упростить этот наиважнейший в области вычислительной техники процесс. Под сортировкой в узком смысле контекста вычислительной техники понимают упорядочивание записей внутри структуры данных.

Актуальность решения технических задач в данной области сохраняется и на сегодняшний день. Дональд Кнут в своей книге «Сортировка и поиск» утверждает, что алгоритмы сортировки занимают половину времени исполнения всех процессов обработки данных вычислительной машиной.

Автор привязывает это к трем основным причинам:

- алгоритмы сортировки находят очень широкое применение;
- алгоритмы сортировки используются часто без надобности;
- используются несовершенные алгоритмы сортировки.

Решение этих проблем является приоритетом в области сортировки с целью ускорения обработки данных в ЭВМ. Создание такой системы, которая использовала бы алгоритмы сортировки тогда и только тогда, когда это требуется, и при этом для определенных типов данных и архитектур ЭВМ применялись бы наиболее подходящие алгоритмы, в значительной степени экономит ресурсы ЭВМ, затрачиваемые на обработку данных, сделает более рациональным использование оперативной памяти при обращении к большим массивам данных. [2]

Отличительной особенностью сортировки является то обстоятельство, что эффективность алгоритмов, реализующих ее, прямо пропорциональна сложности понимания этого алгоритма. Другими словами, чем легче для понимания метод сортировки массива, тем ниже его эффективность.

Самый известный алгоритм — пузырьковая сортировка (bubble sort, сортировка методом пузырька, или просто сортировка пузырьком). Его популярность объясняется интересным названием и простотой самого алгоритма. Тем не менее, в общем случае это один из самых худших алгоритмов сортировки.

Шейкер-сортировка является усовершенствованным методом пузырьковой сортировки.

Анализируя метод пузырьковой сортировки, можно отметить два обстоятельства:

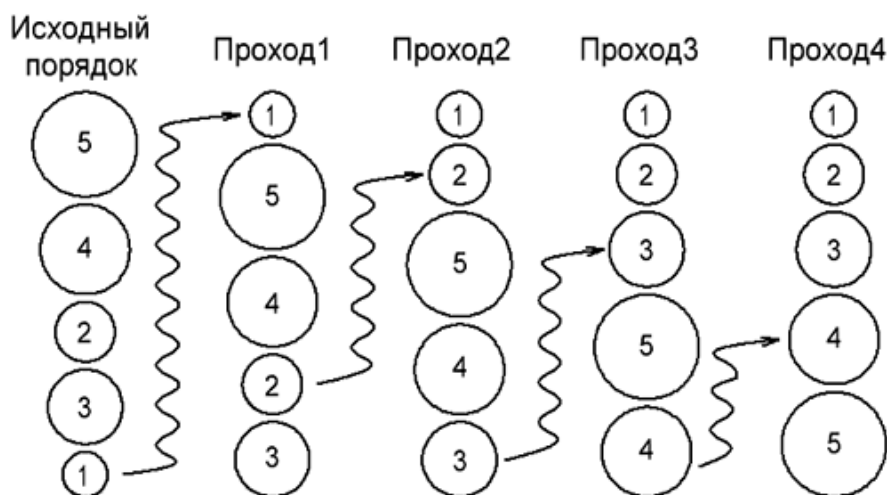
- если при движении по части массива перестановки не происходят, то эта часть массива уже отсортирована и, следовательно, ее можно исключить из рассмотрения;

- при движении от конца массива к началу минимальный элемент "всплывает" на первую позицию, а максимальный элемент сдвигается только на одну позицию вправо. [4]

Эти две идеи приводят к модификациям в методе пузырьковой сортировки.

- от последней перестановки до конца (начала) массива находятся отсортированные элементы. Учитывая данный факт, просмотр осуществляется не до конца (начала) массива, а до конкретной позиции. Границы сортируемой части массива сдвигаются на 1 позицию на каждой итерации;

- массив просматривается поочередно справа налево и слева направо;
- просмотр массива осуществляется до тех пор, пока все элементы не встанут в порядке возрастания (убывания);
- количество просмотров элементов массива определяется моментом упорядочивания его элементов.



Хранение перемещаемого значения осуществляется в буфере памяти. Лучший случай для этой сортировки — отсортированный массив ($O(n)$), худший — отсортированный в обратном порядке ($O(n^2)$). [1]

Анализ алгоритма шейкер-сортировки приводит к следующим наблюдениям:

1. Если при некотором из проходов нет перестановок, то алгоритм можно завершить.

2. Если зафиксировать индекс к последнему обмену (все пары левее этого индекса уже упорядочены), то просмотр можно завершить на этом индексе, а не идти до нижнего предела для индекса [i].

3. Чередование направлений для просмотра (всплывает самый легкий, а тонет самый тяжелый). [3]

Шейкер-сортировка с успехом используется в тех случаях, когда известно, что элементы почти упорядочены.

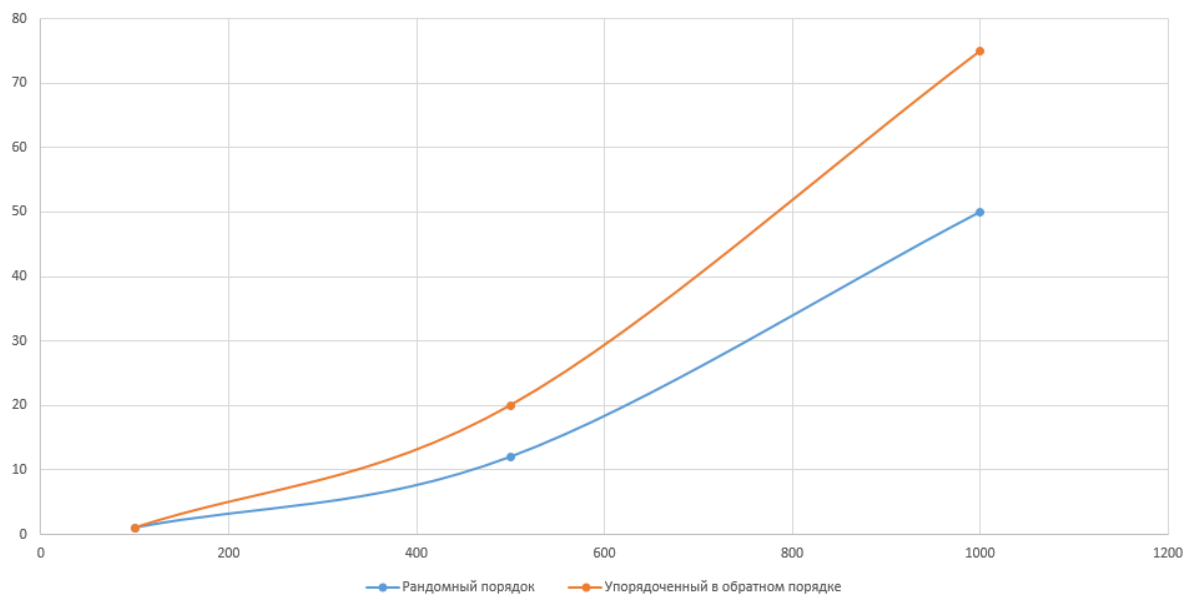
Анализ эффективности шейкер-сортировки можно вывести с помощью общей оценки показателей ускорения и эффективности по двум формулам:

$$S_p = \frac{p \log_2 n}{\log_2(n/p) + 2p}, \quad E_p = \frac{\log_2 n}{\log_2(n/p) + 2p}$$

Проведя анализ по тестированию программы шейкер-сортировки на массивах разной размерности, заполняется таблица зависимости времени от порядка элементов в массиве (время представлено в виде 1/10000 секунд, деление сделано по причине очень быстрой работы сортировки).

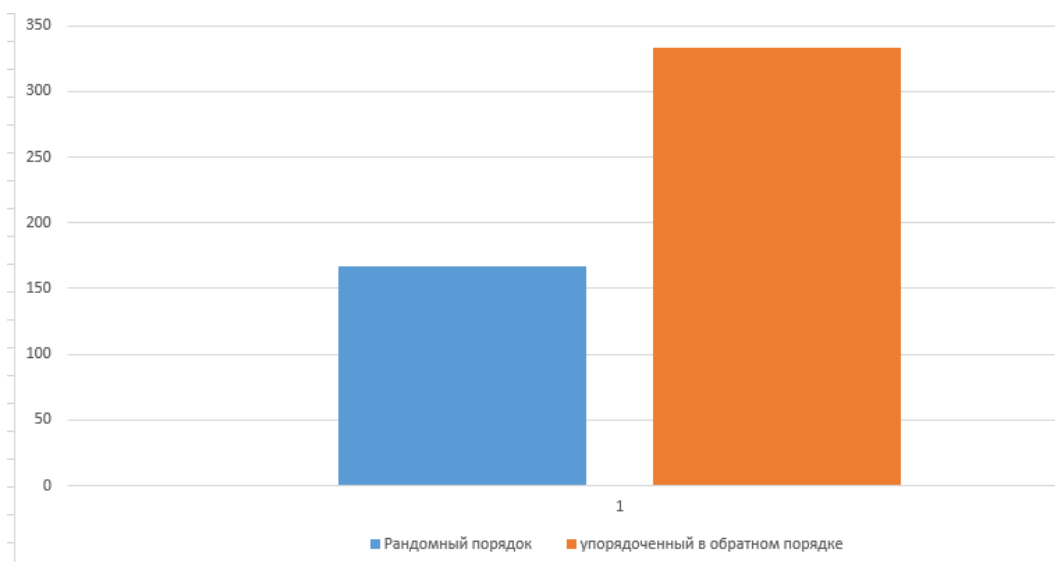
Кол-во -элементов	Массив упорядоченный в случайном порядке	Массив упорядоченный	Массив упорядоченный в обратном порядке
100	1	0,0...1	1
500	12	0,0...1	20
1000	50	0,0...1	75

По приведенной таблице строится график зависимости времени работы шейкер-сортировки от количества элементов в массиве.



По графику можно сделать вывод, что сортировка, упорядоченная в обратном порядке, значительно медленнее, чем сортировка случайных элементов. Так же при увеличении количества элементов разница во времени работы массива упорядоченная в обратном порядке, значительно увеличивается.

Анализ сортировки массивов 100000 элементов заполненным в случайном и упорядоченном в обратном порядке показывает, что случайный набор элементов работает почти в 2 раза быстрее чем в массиве, упорядоченном в обратном порядке, что можно увидеть на диаграмме.



Анализируя шейкер-сортировку в массивах можно сделать вывод, что, время работы шейкер-сортировки напрямую зависит не только от количества элементов в массиве, но и от порядка его заполнения.

Список литературы.

1. Алгоритмы. Построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн. – Москва: ИД «Вильямс», 2013. - 1324 с.
2. Ахо А.В., Хопкрофт Д., Ульман Дж.Д. Структуры данных и алгоритмы / Пер. с англ. Минько А.А. – Москва: ИД «Вильямс», 2000. - 384 с.
3. Гудман С. Введение в разработку и анализ алгоритмов / Гудман С., Хидетниемеи С. – Москва: Мир, 1981. – 368с.
4. Д.Кнут. Искусство программирования для ЭВМ. Сортировка и поиск / Д.Кнут. – Москва: Мир, 1978. – 355 с.
5. Никлаус Вирт. Алгоритмы и структуры данных. Новая версия для Оберона / Пер. с англ. Ткачев Ф. В. – Москва: ДМК Пресс, 2010. – 272 с.
6. Роберт Седжвик. Фундаментальные алгоритмы на С++. Части 1-4. Анализ. Структуры данных. Сортировка. Поиск / Р. Седжвик. – Киев: ДиаСофт, 2002. – 688 с.