

Сортировка по ДСТУ 9101

Быстрая сортировка

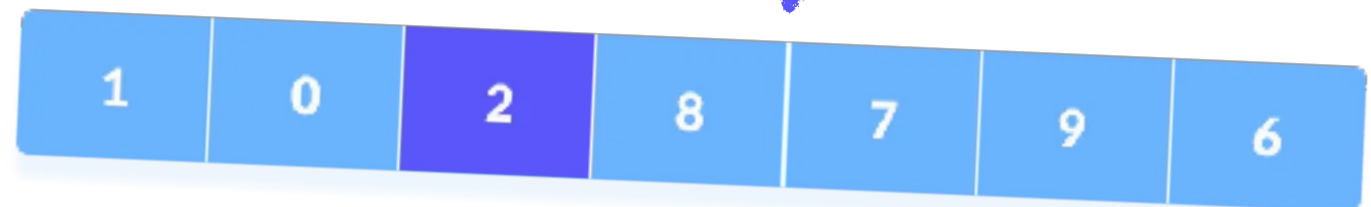
Петрищев Т. В.
Шкодских В. В.
Ткач В. Д.
Бодунов В. А.
Нгуен В.

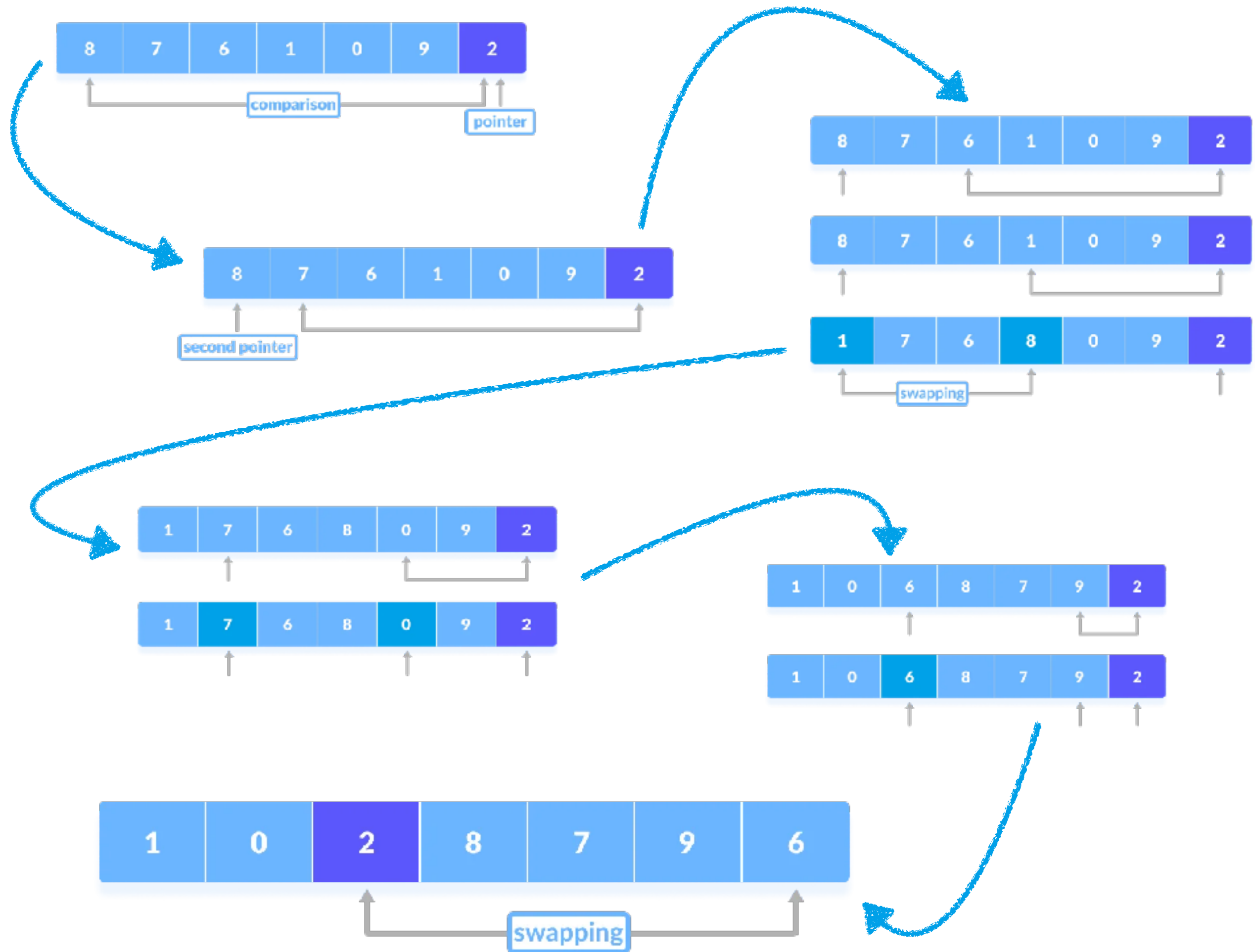
Чарльз Хоар

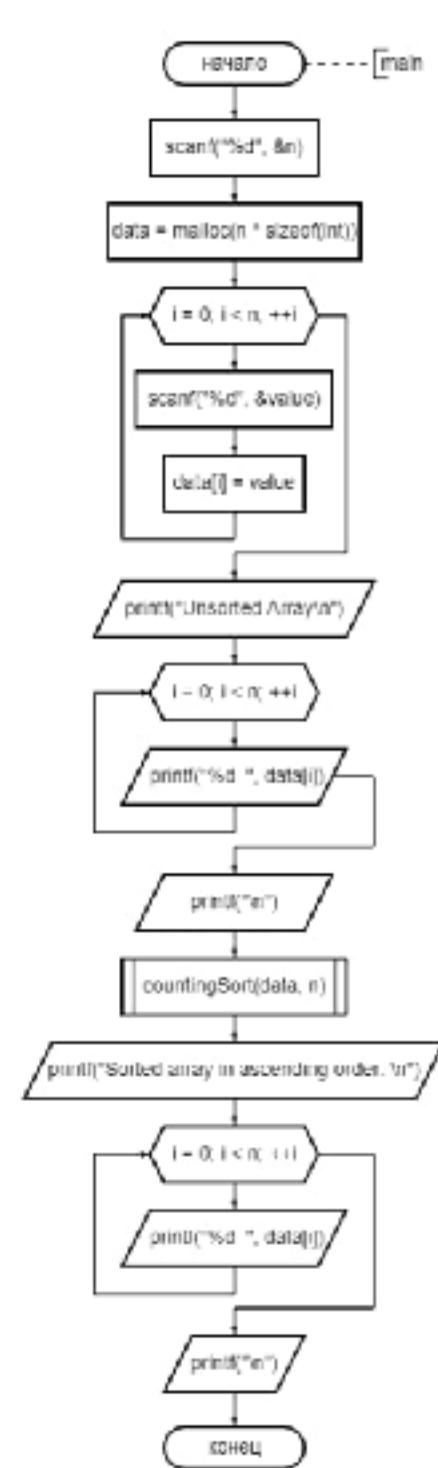
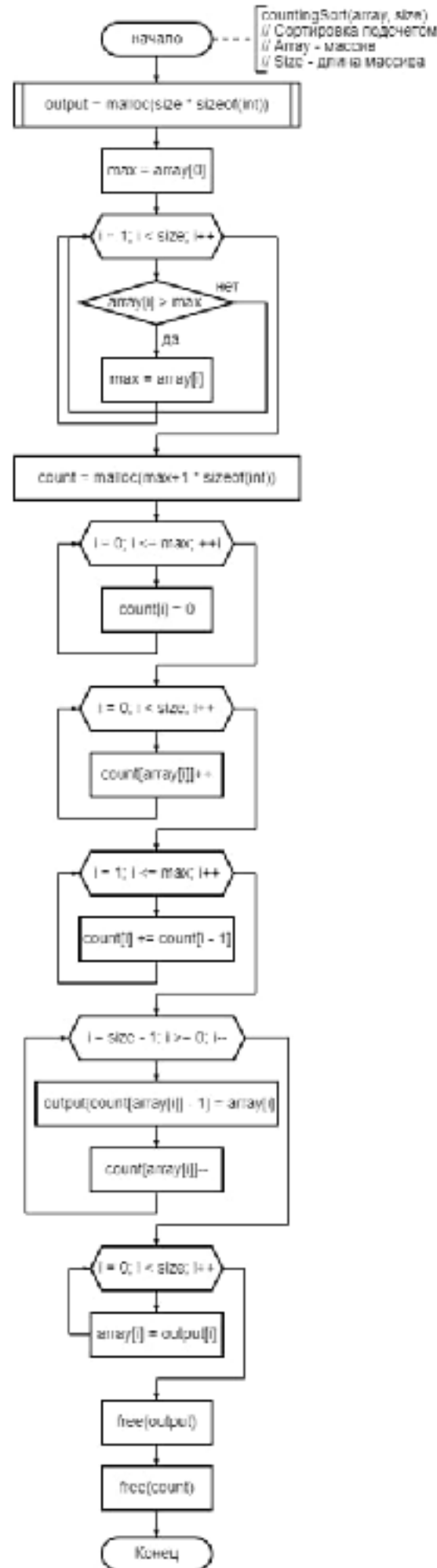


Quick Sort

– это алгоритм сортировки,
основанный на принципе «разделяй и властвуй».







Дефективность

✓ Самый быстрый
в среднем случае

По времени:

Лучшая – $O(n * \log n)$

Худшая – $O(n^2)$

Средняя – $O(n * \log n)$

По памяти: $O(\log n)$

✗ Очень плохая
производительность в
худшем случае

✗ Необходимо
дополнительное
стековое пространство
 $O(\log n)$

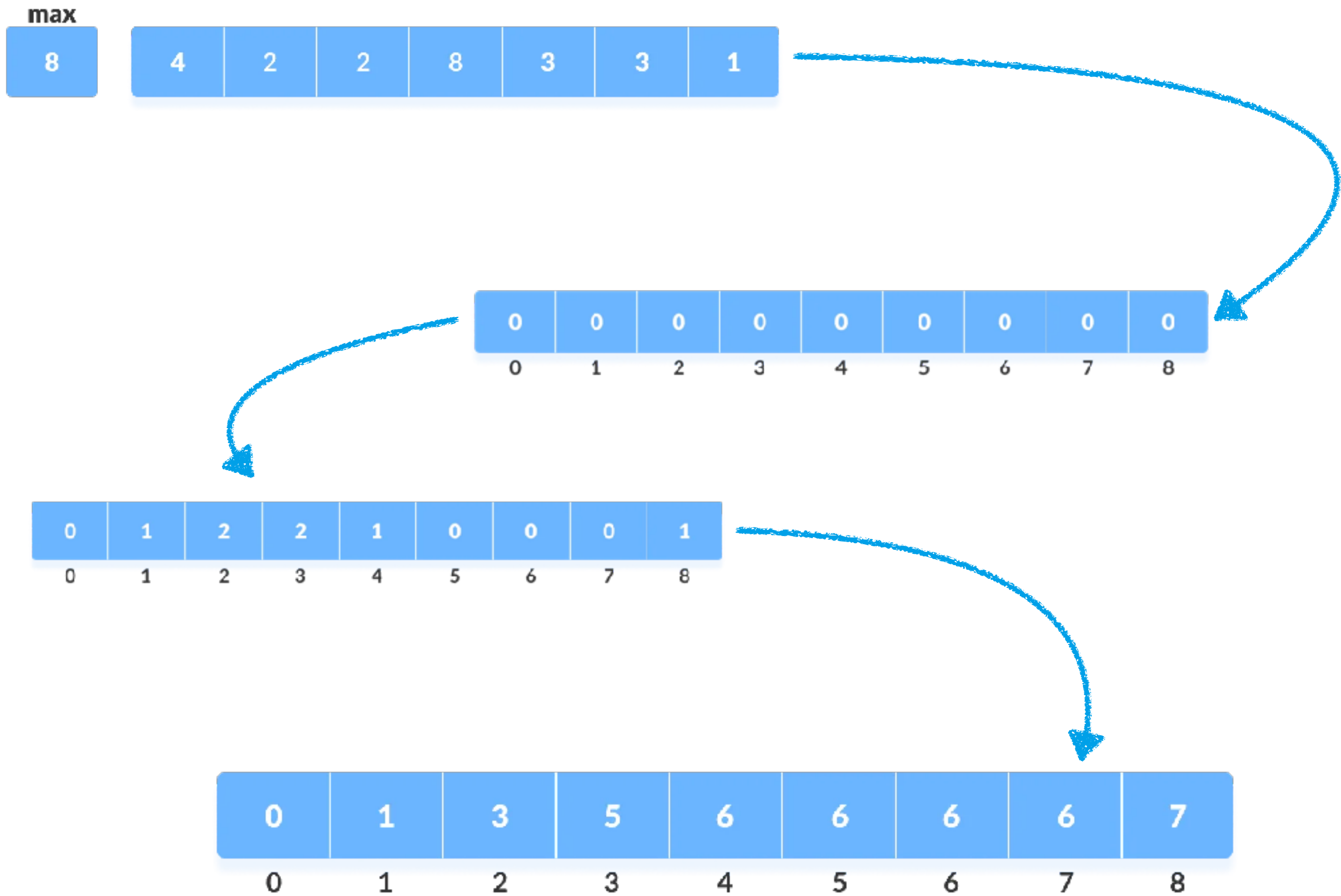
✗ Нестабильная
сортировка

Counting Sort



– сортирует
элементы массива, подсчитывая количество вхождений
каждого уникального элемента в массиве. Счетчик
хранится во вспомогательном массиве, а сортировка
выполняется путем отображения счетчика как индекса
вспомогательного массива.

Гарольд Х. Сьюард



array

4	2	2	8	3	3	1
---	---	---	---	---	---	---

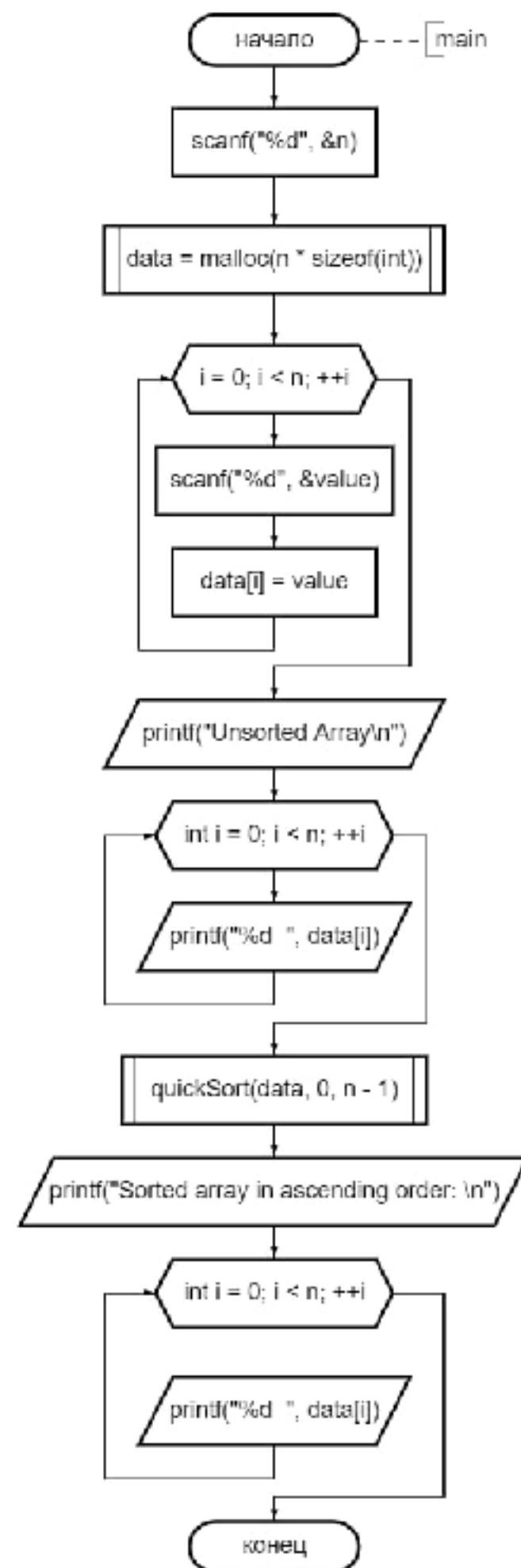
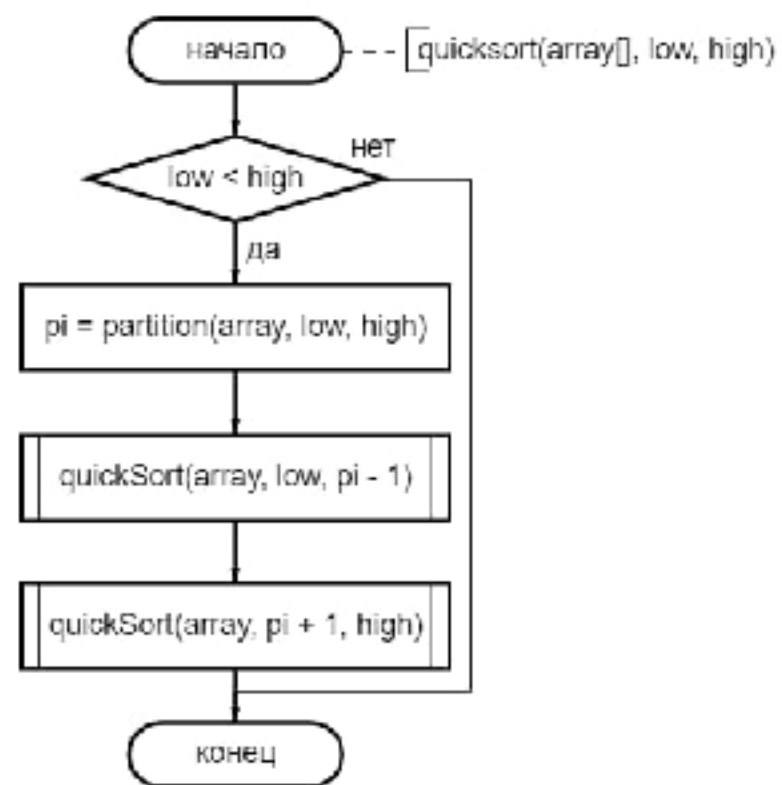
count

0	1	2	3	4	5	6	7	8
0	1	3	5	6	6	6	6	7

$$6 - 1 = 5$$

output

0	1	2	3	4	5	6
1	2	2	3	3	4	8





✓ Очень быстро
работает, если разброс
входных значений
невелик

По времени:

Лучшая – $O(n + k)$

Худшая – $O(n + k)$

Средняя – $O(n + k)$

По памяти: $O(max)$

✗ Работает медленно,
если диапазон
значений $> \log(n)$

✗ Требуется
дополнительная
память

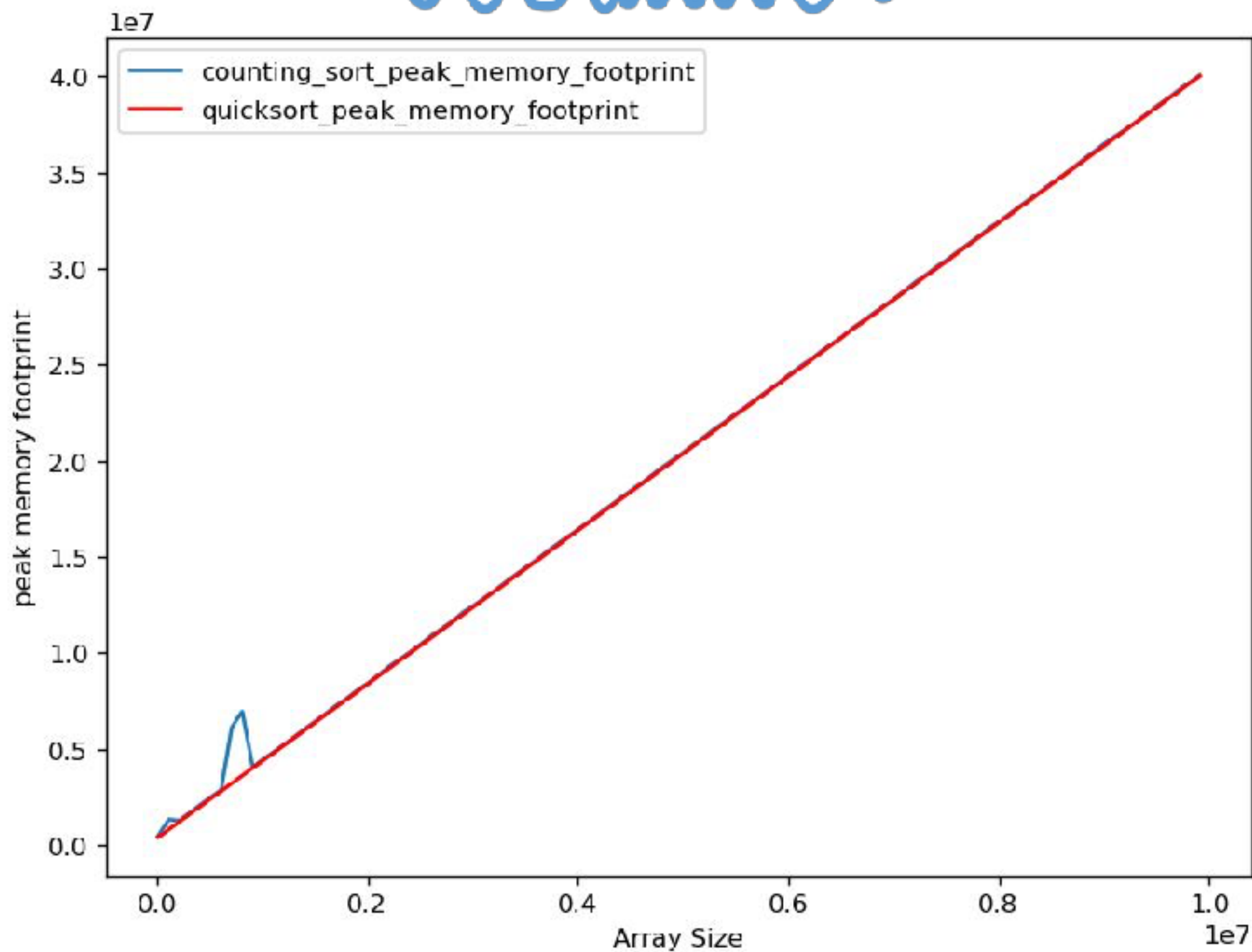
✗ Работает только с
данными целого типа

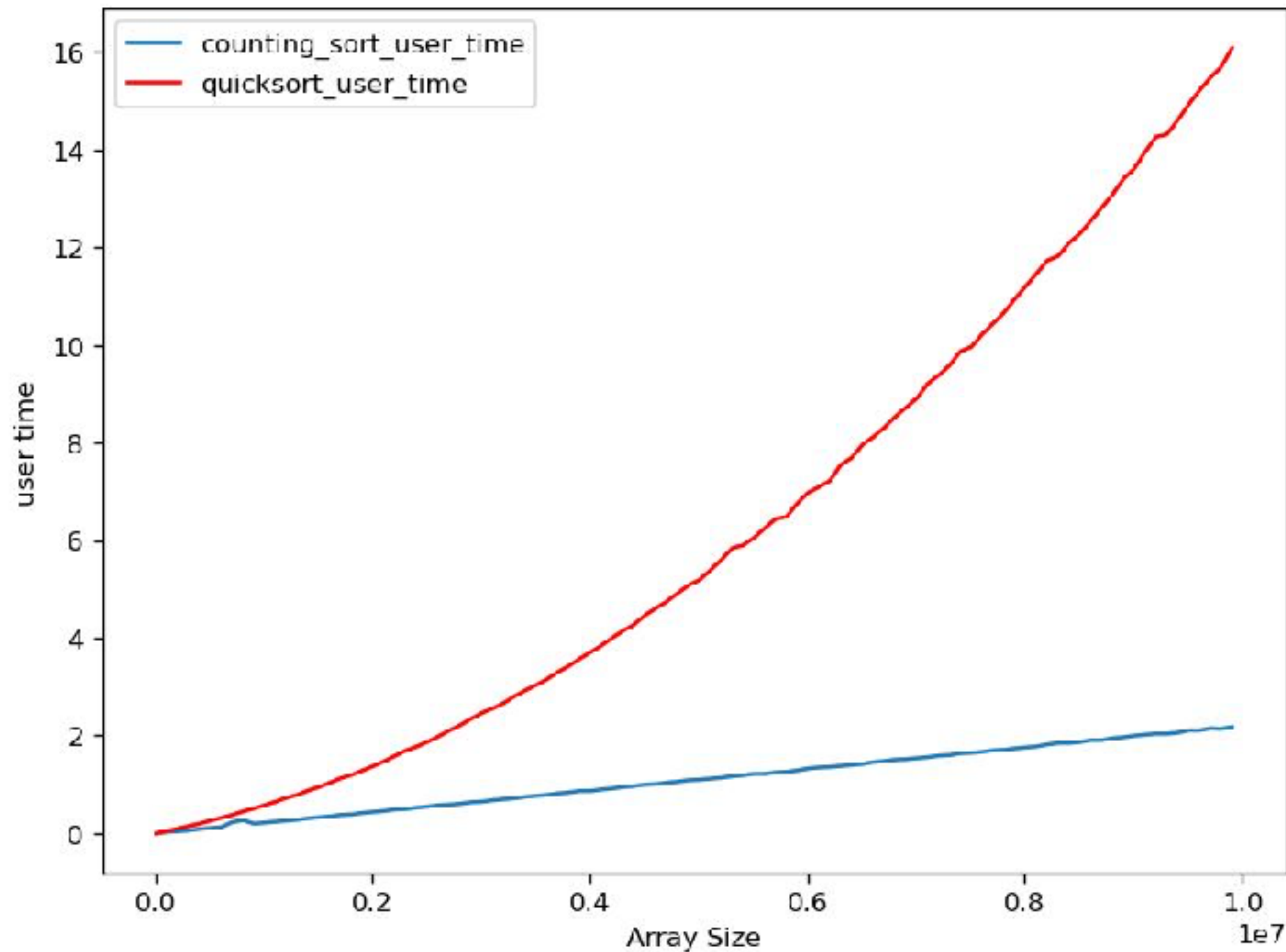
ГРАФИКИ!



Кривая нормального распределения

Использование памяти





Затраты по времени

Спасибо за внимание.

Источники

- <https://www.programiz.com/dsa/counting-sort>
- <https://www.programiz.com/dsa/quick-sort>
- <https://www.adamk.org/quick-sort/>
- <https://habr.com/ru/company/otus/blog/524948/>
- <https://prog-cpp.ru/sort-fast/>
- www.algolab.valemak.com/counting
- https://spravochnick.ru/informatika/sortirovka_podschetom
- <https://vscode.ru/prog-lessons/sortirovka-metodom-podscheta-nasi.html>