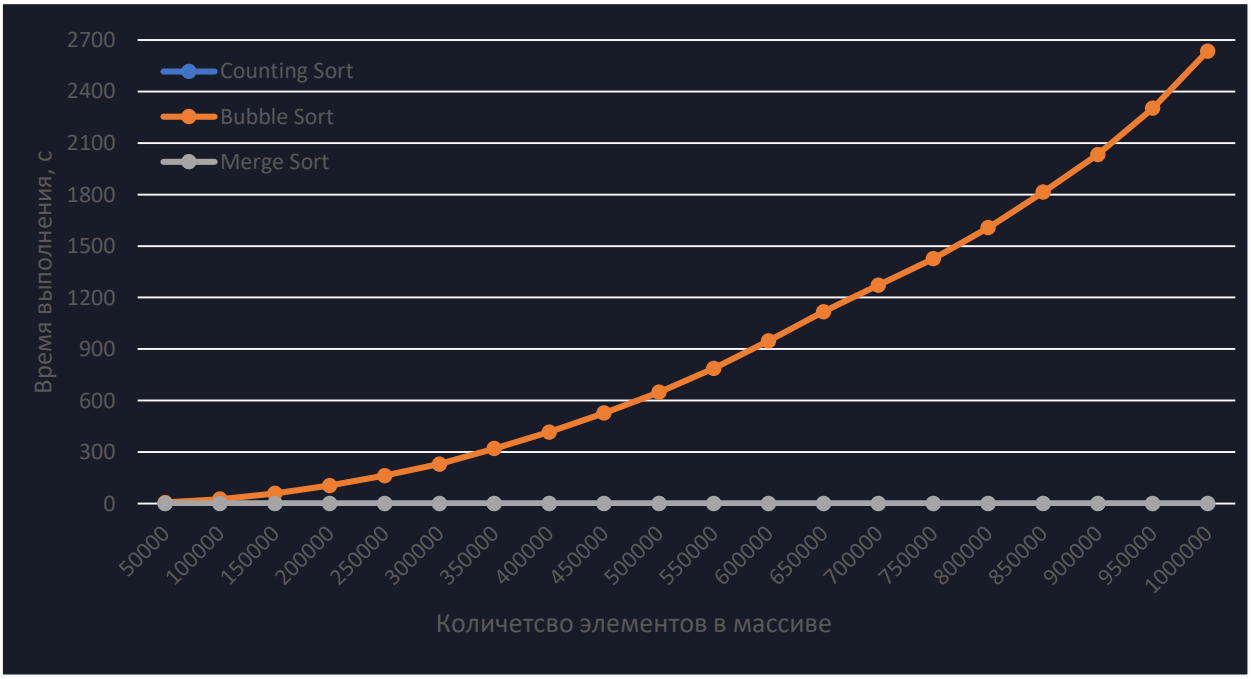
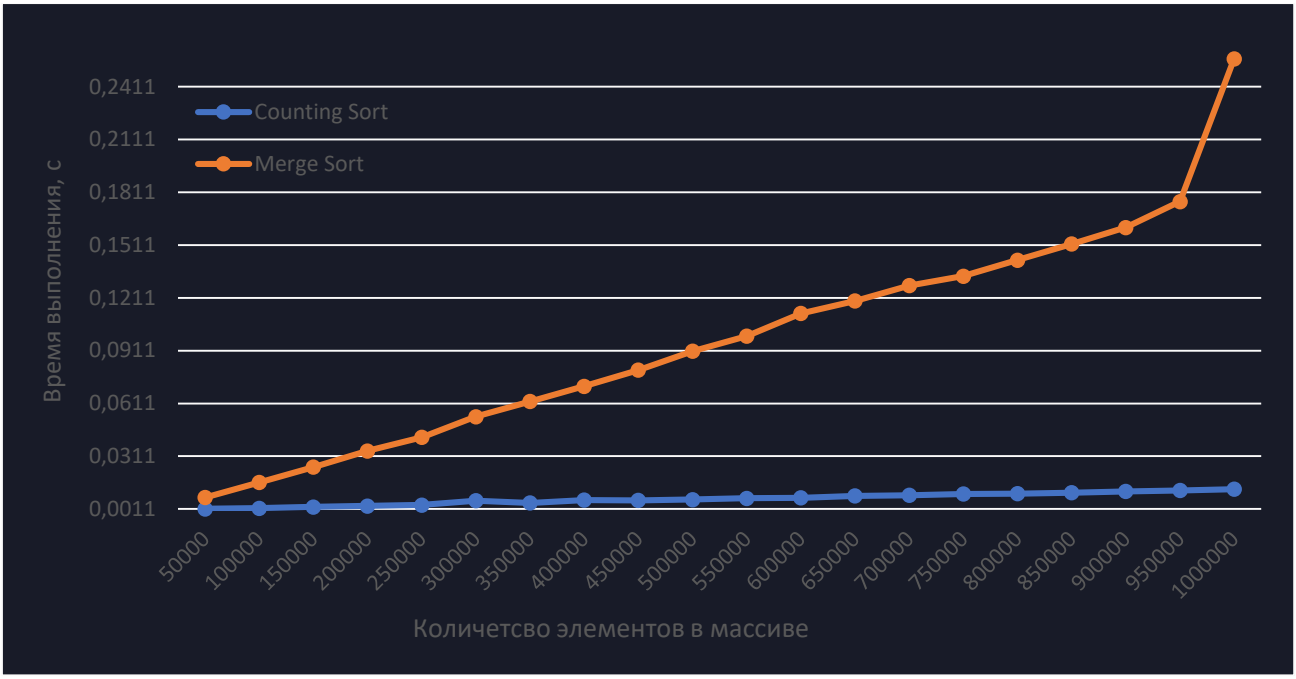
#	Количество элементов в массиве	Counting Sort, с.	Bubble Sort, с.	Merge Sort, с.
1	50000	0,001166	6,296584	0,007664
2	100000	0,001594	25,943993	0,016188
3	150000	0,002244	59,039106	0,024895
4	200000	0,002819	105,298140	0,034085
5	250000	0,003402	162,729235	0,041872
6	300000	0,005796	230,011327	0,053553
7	350000	0,004570	320,557326	0,062273
8	400000	0,006164	416,149748	0,070847
9	450000	0,006017	526,689409	0,080042
10	500000	0,006442	648,248516	0,090777
11	550000	0,007238	786,782345	0,099320
12	600000	0,007485	947,850551	0,112274
13	650000	0,008582	1116,756656	0,119407
14	700000	0,008887	1271,920310	0,128132
15	750000	0,009642	1426,059830	0,133501
16	800000	0,009802	1607,080622	0,142534
17	850000	0,010396	1814,136469	0,151794
18	900000	0,011104	2032,912061	0,161147
19	950000	0,011614	2301,505320	0,175843
20	1000000	0,012399	2634,563352	0,257009



«Зависимость времени выполнения алгоритмов Counting Sort, Bubble Sort, Merge Sort от размера массива»



«Зависимость времени выполнения алгоритмов Counting Sort, Merge Sort от размера массива»

Контрольные вопросы

1. Вычислительная сложность алгоритма – это количество вычислительных ресурсов, необходимых для выполнения алгоритма, как правило, измеряется в количестве операций.

2. f(n) = O(g(n)) означает, что f(n) растет не быстрее, чем g(n), f(n) = (Θ g(n)) означает, что f(n) растет пропорционально g(n), f(n) = Ω(g(n)) означает, что f(n) растет не медленнее, чем g(n).

3. Устойчивый (stable) алгоритм сортировки сохраняет относительный порядок элементов с одинаковыми ключами в отсортированном массиве.

4. Алгоритм сортировки "на месте" (in-place) не требует дополнительной памяти для хранения временных данных при сортировке.

5. Вычислительная сложность в худшем случае:
Сортировка Counting Sort – O(n+k)

Сортировка Counting Sort основывается на подсчете количества элементов каждого значения в массиве, и затем на создании отсортированного массива, используя эти подсчеты. Сложность алгоритма определяется временем, необходимым для подсчета количества элементов и временем, необходимым для создания отсортированного массива. Время подсчета количества элементов занимает O(n + k) времени, а время создания отсортированного массива занимает O(n) времени.

Однако, необходимо учитывать, что Counting Sort имеет ограничение на диапазон значений элементов, так как требует создания массива для подсчета количества элементов каждого значения. Если диапазон значений слишком велик, то это может привести к высокому расходу памяти и ухудшению производительности.

Сортировка Bubble Sort – O(n^2)

Алгоритм сортировки пузырьком работает путем сравнения каждой пары соседних элементов массива и, при необходимости, переставляет их местами. Он продолжает проходы по массиву до тех пор, пока не будет отсортирован весь массив.

Хотя алгоритм прост в реализации и легко понятен, его эффективность ограничена, особенно при сортировке больших массивов. В среднем и лучшем случаях, когда массив уже отсортирован или почти отсортирован, сложность может быть улучшена до O(n), но в худшем случае, когда массив находится в обратном порядке, количество операций сравнения и перемещения элементов может быть в квадрате от количества элементов в массиве, что делает этот алгоритм неэффективным для больших массивов.

Сортировка Merge Sort – O(n log(n))

Алгоритм сортировки слиянием работает путем разделения исходного массива на две части, сортировки каждой части отдельно, а затем объединения отсортированных частей в один отсортированный массив. Для сортировки каждой части используется рекурсия.

Сложность сортировки слиянием обусловлена количеством сравнений, необходимых для слияния двух отсортированных массивов. Количество сравнений для слияния двух массивов размера n/2 каждый составляет O(n), что приводит к общей сложности O(n log n) для сортировки всего массива.

Хотя Merge Sort имеет более высокую сложность, чем некоторые другие алгоритмы сортировки, такие как Quick Sort, он гарантирует стабильную и предсказуемую производительность во всех случаях, включая худший. Кроме того, Merge Sort имеет дополнительные преимущества, такие как устойчивость к наихудшему случаю и возможность параллельной обработки, что делает его эффективным для сортировки больших массивов данных.

6. На графиках мы можем наблюдать, что алгоритмы Counting и Merge работают намного быстрее алгоритма Bubble на всех размерах входных данных. Поэтому кривые у алгоритмов Counting и Merge более плавные (не смотря на переход от 950000 до 1000000 у алгоритма Merge Sort), что говорит о лучшей вычислительной сложности алгоритмов по сравнению с алгоритмом Bubble Sort. Вычислительная сложность у алгоритма Merge Sort

– O(n log(n)), а у Counting Sort – O(n+k). Bubble Sort имеет вычислительную сложность O(n^2), из-за чего кривая на графике более крутая

7. Алгоритмы сортировки с вычислительной сложностью O(n log n) для худшего случая: сортировка слиянием(Merge Sort), пирамидальная сортировка(Heap Sort), Quick Sort и т. д..

8. Сортировка Radix sort - O(nk).