

Диаграмма для первого пункта

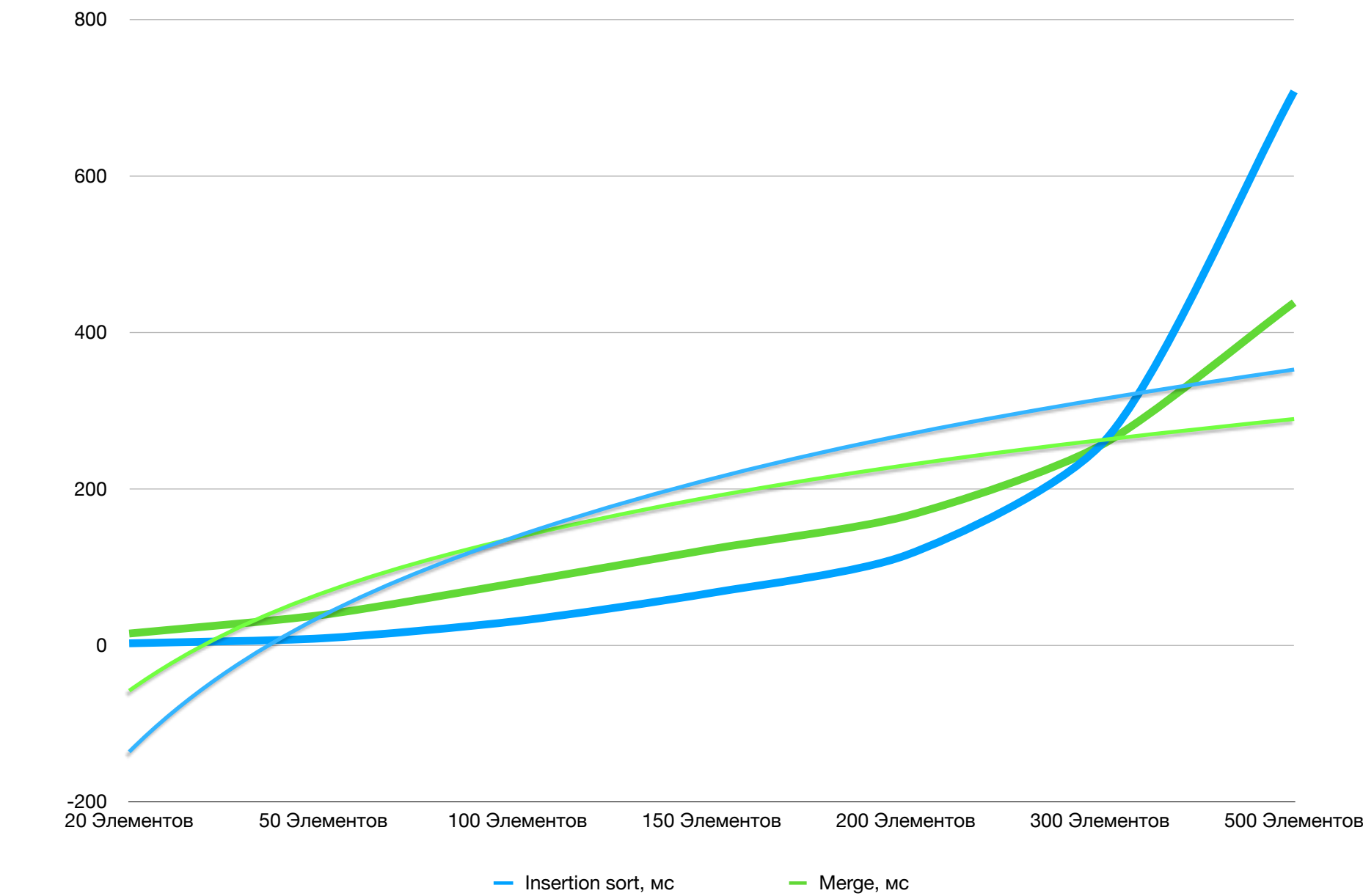


Таблица для первого пункта

	Insertion sort, mc	Merge, mc
20 Элементов	2	14
50 Элементов	8	38
100 Элементов	30	79
150 Элементов	66	122
200 Элементов	114	164
300 Элементов	254	254
500 Элементов	707	437

Как мы видим для числа элементов до 300 сортировка вставками работает быстрее, а на совсем маленьких данных даже сильно быстрее чем сортировка слиянием, но после этого сортировка слиянием начинает работать гораздо быстрее сортировки вставками, а также видно, что линия тренда также устремляется к графику сортировки слиянием, что говорит о том, что она на $n > 10^6$ будет работать эффективнее

Q sort Merge sort Heap sort Guaranteed Qsort Skip list sort

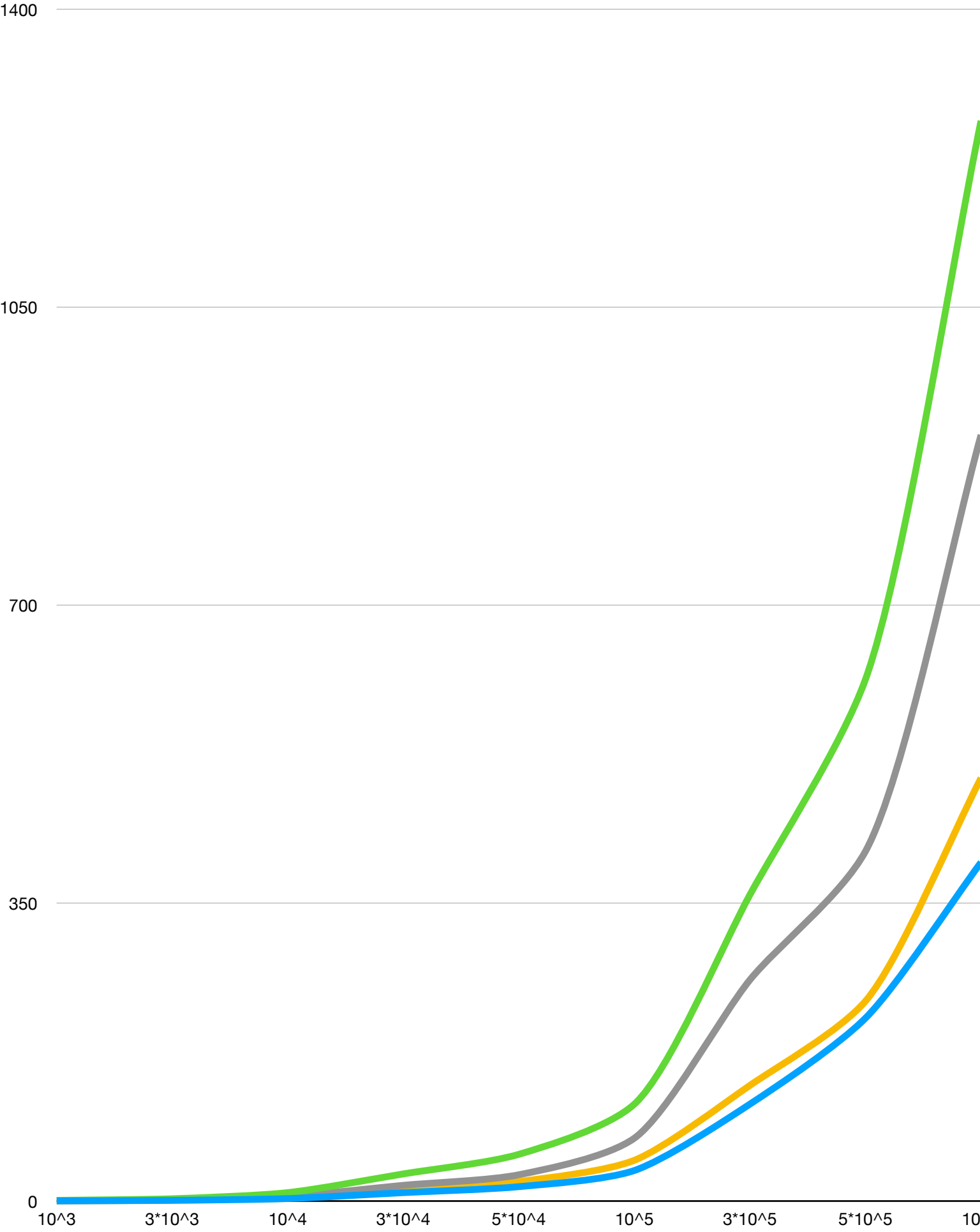


Таблица для быстрых сортировок

Число Элемнтов / виды сортировки	Merge sort	Q sort	Heap sort	Guaranteed Qsort	Skip list sort
10^3	0,9	0,3	0,4	0,3	
3*10^3	2,8	0,9	1,4	1,2	
10^4	10	3	5,5	3,5	
3*10^4	32	10	18,6	13	
5*10^4	55	17	31	23	
10^5	114	36	74	48	
3*10^5	360	114	260	136	
5*10^5	613	215	411	235	
10^6	1269	398	900	497	

Как видно из диаграмм, лучше всего работает qsort, после него идет Гарантированный метод qsort, но важно понимать, что тестирования проводятся на рандомно сгенерированных массивах, поэтому потом проведем отдельные тестирования для этих двух сортировок выполненных на массивах отсортированных в обратном порядке, чтобы показать почему гарантированная сортировка на самом деле лучше. Далее по времени идет Heap sort, который работает хуже чем две описанные сортировки, но все еще лучше чем сортировка слиянием, она работает хуже всех остальных

Таблица для отсортированных в обратном порядке массивов

	Qsort	Guar Qsort
1000	255334	117709
10000	2674125	1207459
100000	29853917	15805042
1000000	294561500	168756042

Как видно из таблицы на данных отсортированных в обратном порядке гарантированная сортировка работает в среднем в 2 раза быстрее обычной

Дабы не быть голословным я провел еще одно маленькое исследование, чтобы показать, что на данных, которые мы знаем, что заведомо плохо отсортированы гарантированная сортировка работает куда лучше чем обычная, приведу статистику в таблице

Таблица для сортировок строк

Число строк/ длины строк	1	10	100
20	0,71	0,73	1,14
200	0,17	0,24	0,36
1200	0,08	0,1	0,17

Заметим, что на 20 строках элементе обычная сортировка вставками примерно также, как и сортировка с бинпоиском, а когда длина строки 100, то и вовсе работает быстрее. Сортировка с бинпоиском работает стабильно быстрее при чем сортировке с бинпоиском, а также с увеличением числа строк в массиве сортировка с бинпоиском сортировка с бинпоиском работает стабильно быстрее чем обычная, но с увеличением числа символов в строке сортировка с бинарным поиском работает дольше чем при строках, длина которых равна 1

потом растет дольше тем при строчках длины которых равен 1, потому что присваивания начинают выполняться дольше

Число строк/ длины строк	1	10	100
3000	0,07	0,08	0,14
10000	0,06	0,07	0,12