

## Анализ результатов работы с сортировками

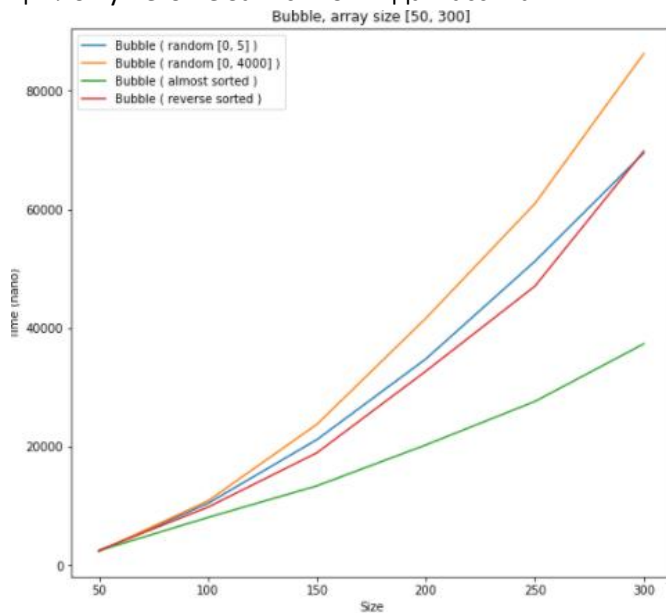
Полный перечень графиков представлен в `sortComp.html` (из jupyter). В начале отображены графики по типам массива, далее – по типам сортировок.

Общий вывод:

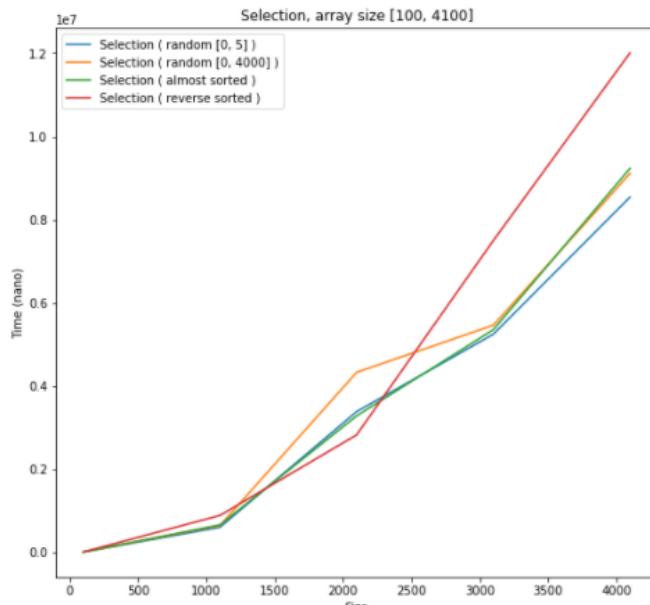
На графиках по типам массивов видно, что пузырьек, выбор, слияние и во многих случаях вставка (однако на почти отсортированном массиве она работает быстрее всех) работают значительно дольше с любыми типами и размерами массивов. В остальном на всех типах массивов, кроме очень близких к отсортированным (тут побеждает вставка), на небольших размерах побеждает быстрая сортировка, а на больших – подсчет.

Некоторые отдельно замеченные нюансы:

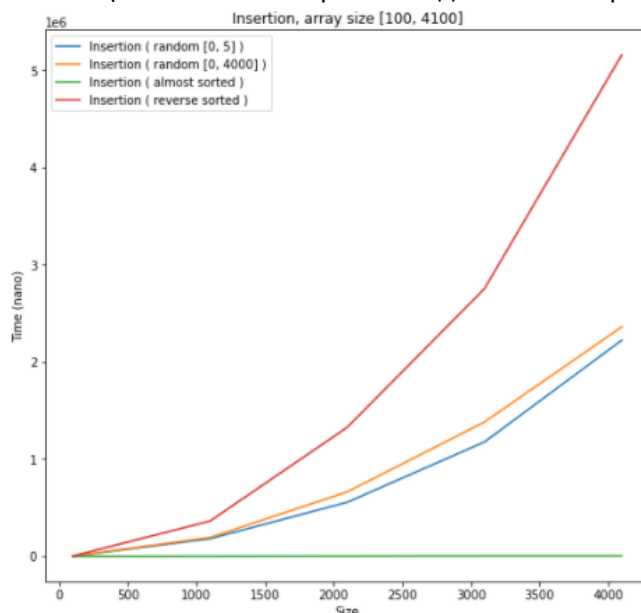
- 1) Пузырёк работает примерно одинаково на всех типах и размерах массива и время работы квадратично зависит от размера массива (график идет по параболе), так как количество циклов у него не зависит от вида массива.



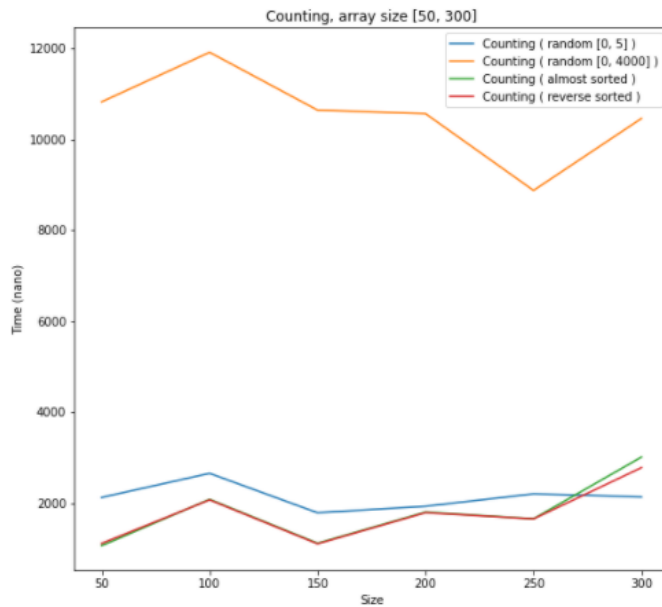
- 2) Сортировка выбором работает немного хуже на обратном отсортированном массиве большой размерности, так как каждый раз ей приходится вплоть до последнего элемента в массиве переприсваивать найденный минимум.



- 3) Сортировка вставкой работает в разы быстрее на почти отсортированном массиве любой размерности, так как количество циклов в ней очень сильно зависит от количества перестановок, так что в полностью отсортированном массиве данный алгоритм отработается за линию. (Соответственно работает дольше на обратно отсортированном).



- 4) Сортировка подсчетом работает значительно дольше в случае, когда разница между максимальным и минимальным элементом сильно превышает количество элементов. В данном случае цикл по вспомогательному массиву будет занимать большую часть времени. При увеличении размера массива или уменьшении разброса чисел сортировка становится эффективнее.



- 5) В быстрой сортировке обратно сортированный массив работает так же быстро, как почти сортированный. Это происходит из-за того, что на самом первом рекурсивном вызове алгоритм полностью переворачивает обратно сортированный массив и все остальные вызовы будут происходить уже на сортированных подчастях.

