

ОТЧЁТ

Подготовка исследования и его этапы

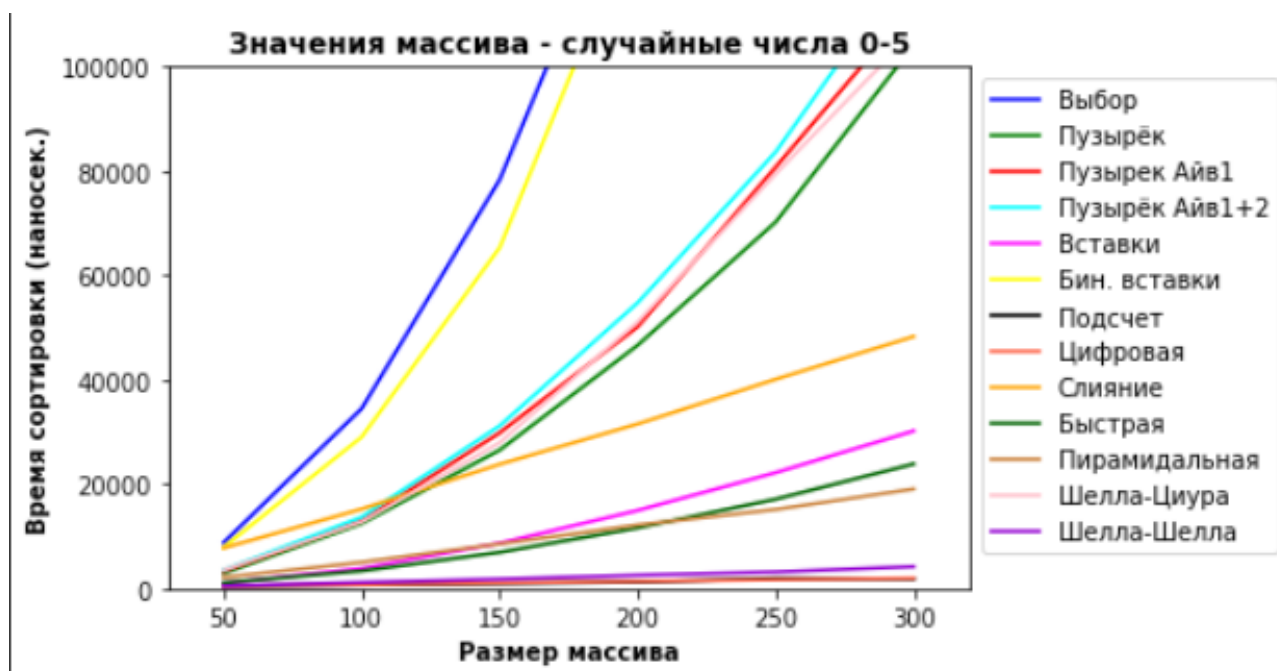
Для проведения экспериментального исследования 13 указанных сортировок в рамках контрольного домашнего задания по предмету "Построение и анализ алгоритмов" было сделано следующее:

- Написаны алгоритмы сортировок
- Составлены 4 вида массивов в соответствии с заданием
- Проведены тесты времени и количества элементарных операций при сортировке массива различными способами
- Результаты тестирования были сохранены в два .csv файла, каждый для своего типа задания (первый - размеры массивов в рамках [50; 300] с шагом в 50; второй - размеры массивов в рамках [100; 4100] с шагом в 100)
- Данные из .csv файлов были импортированы в Jupyter notebook в качестве датафреймов библиотеки pandas
- Были построены графики, отражающие результаты экспериментов, в соответствии с условием задания
- Почти каждый график был прокомментирован
- Был составлен данный отчёт

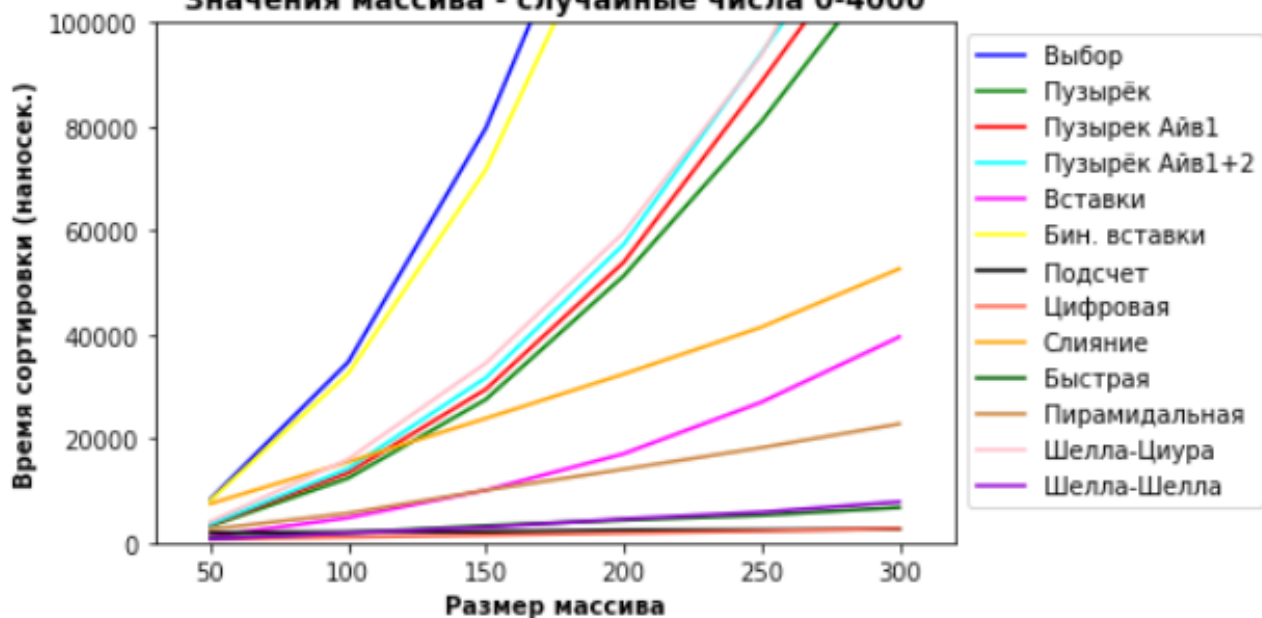
Ключевые моменты, увиденные на графиках

Прежде всего, оценочная асимптотика графиков была подтверждена построенными графиками. В частности, стоит рассмотреть наиболее интересные случаи, которые демонстрируют отличия сортировок на данных массивах.

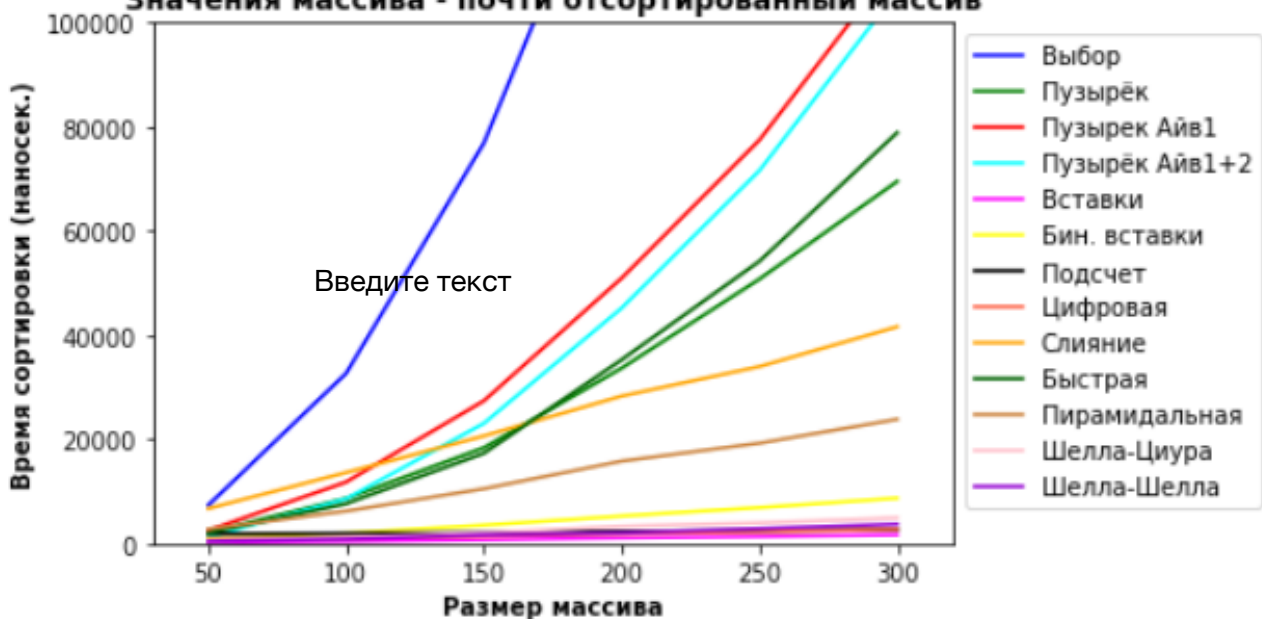
- На малых входных данных (то есть на малых размерах массивов) некоторые квадратичные сортировки работают быстрее, чем сортировки с асимптотикой $O(N \cdot \log N)$. Так, например, сортировка Шелла с последовательностью Шелла и сортировка вставками работают быстрее сортировки слиянием:



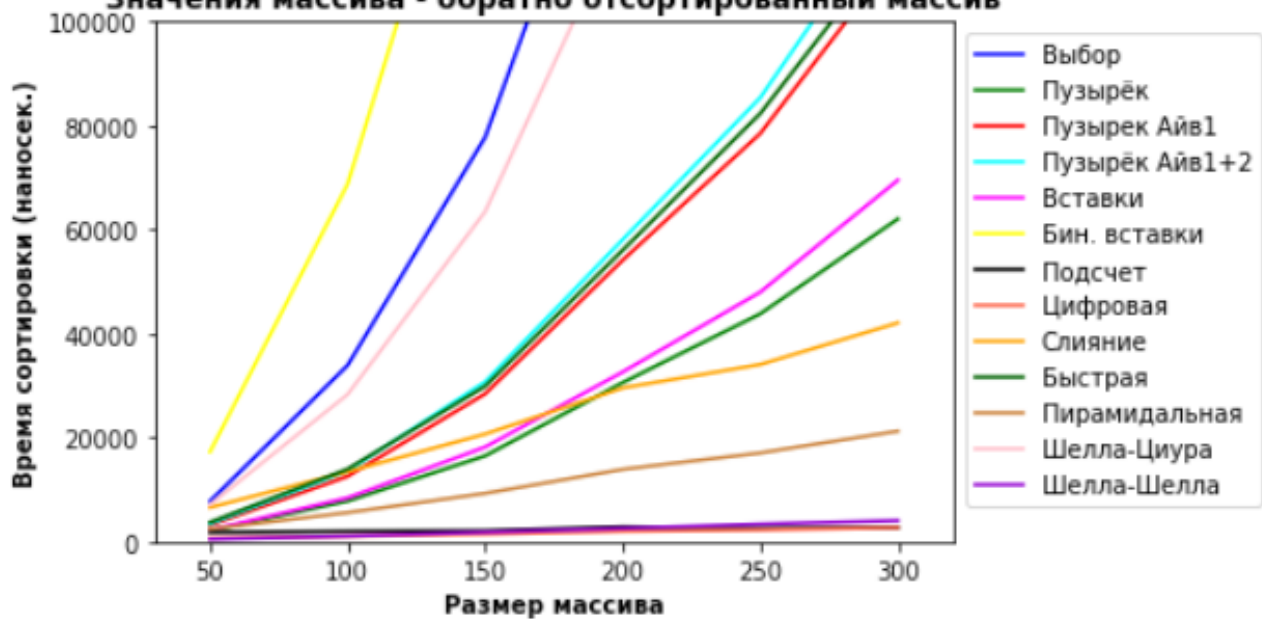
Значения массива - случайные числа 0-4000



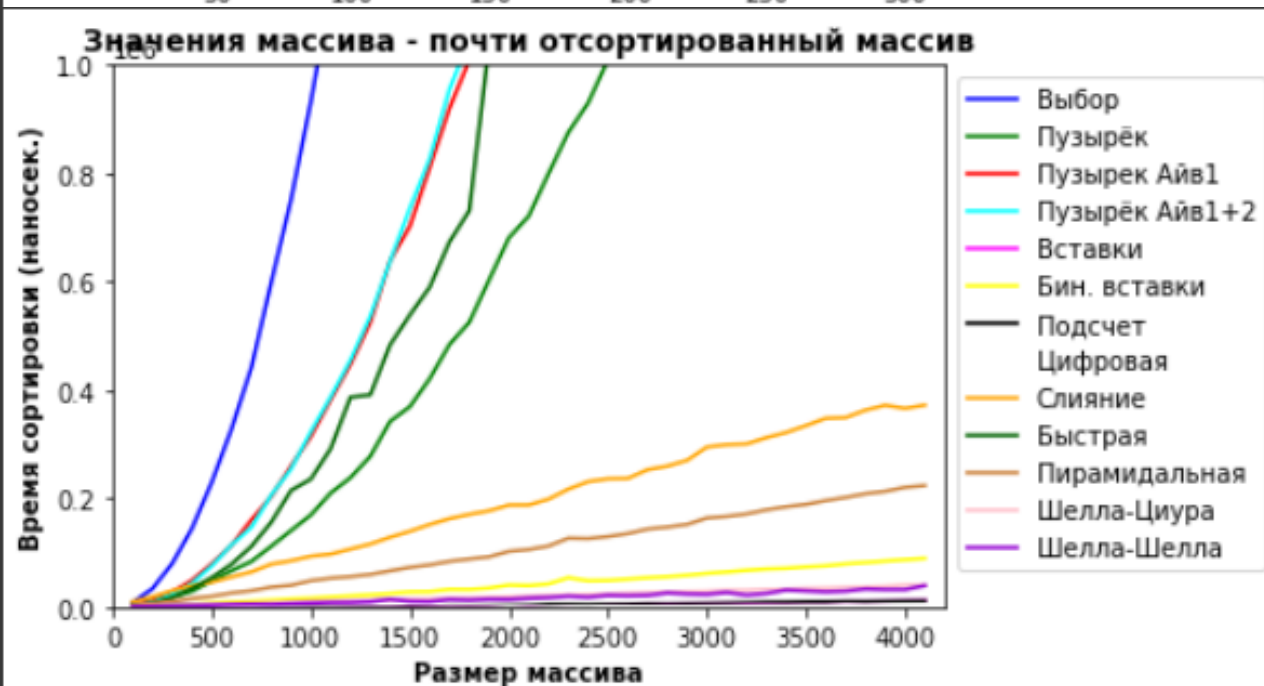
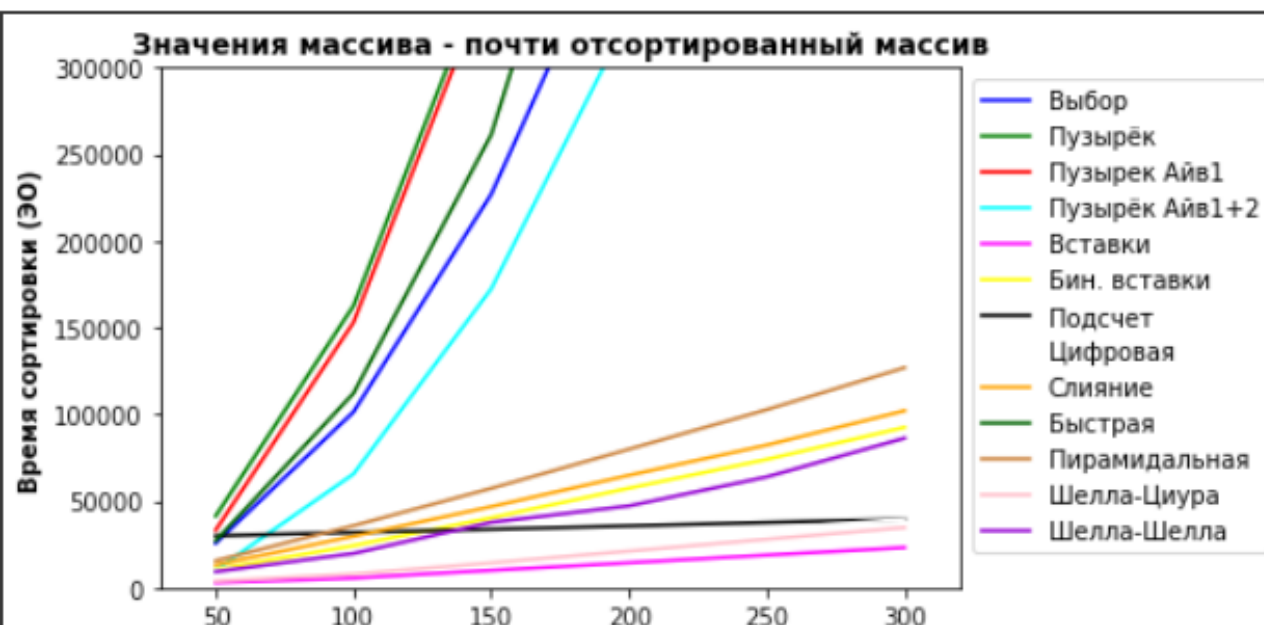
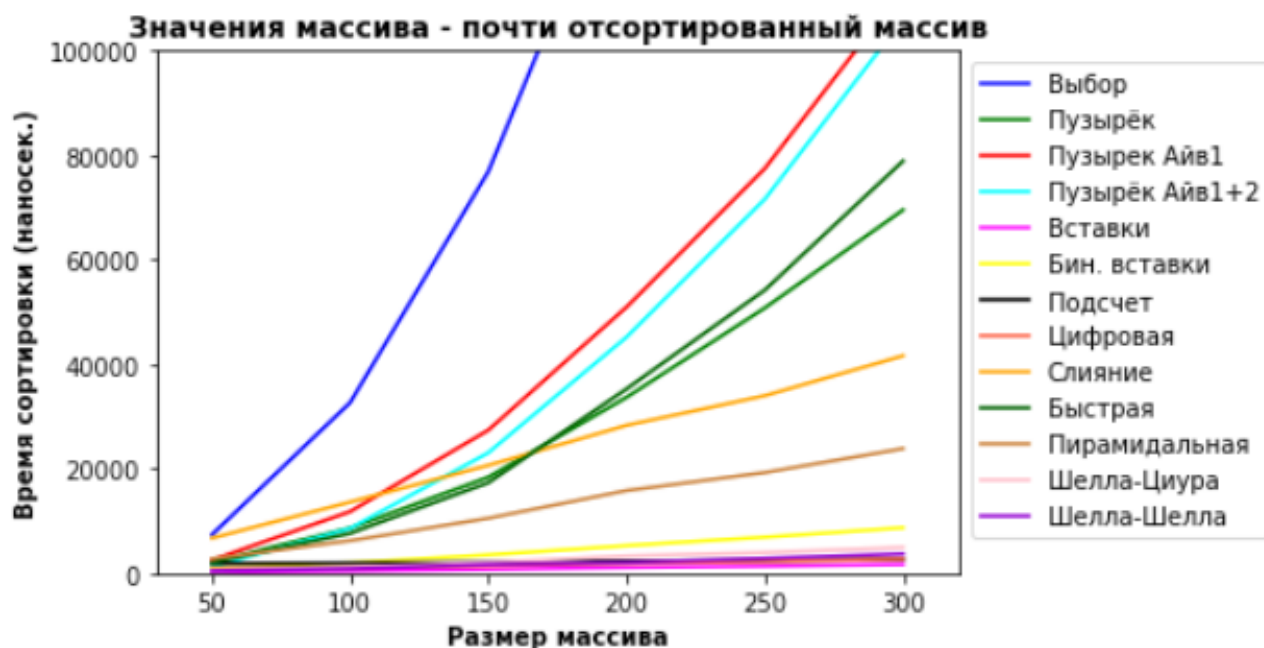
Значения массива - почти отсортированный массив

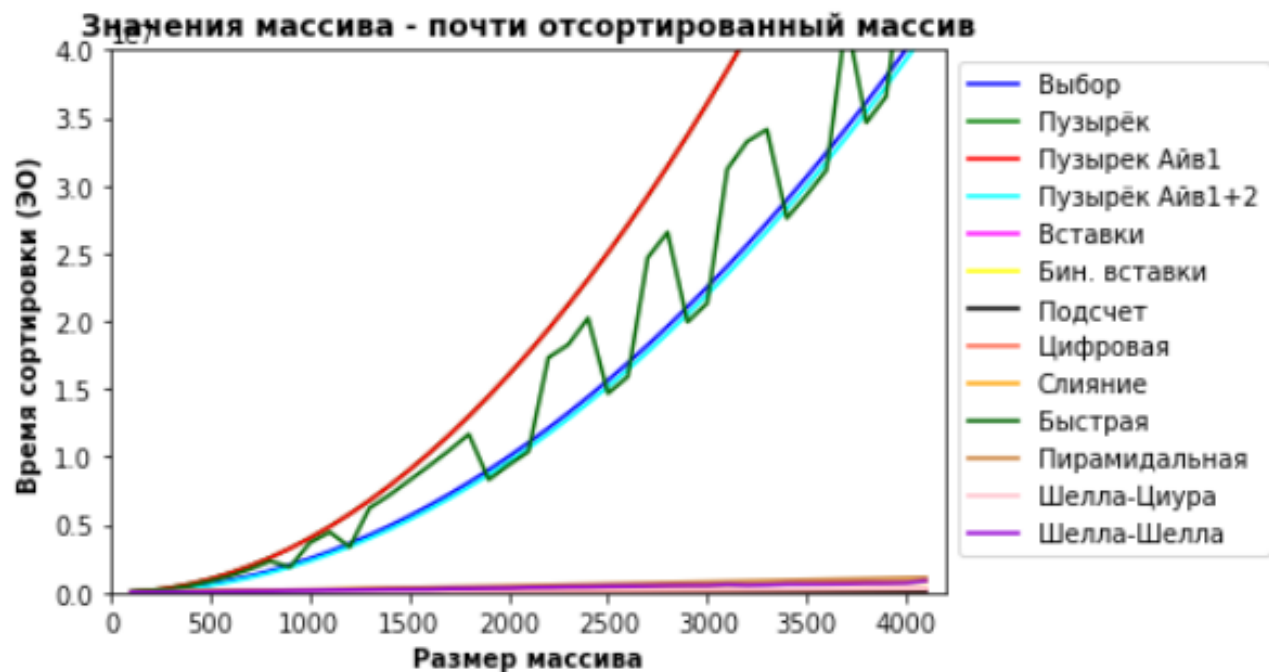


Значения массива - обратно отсортированный массив

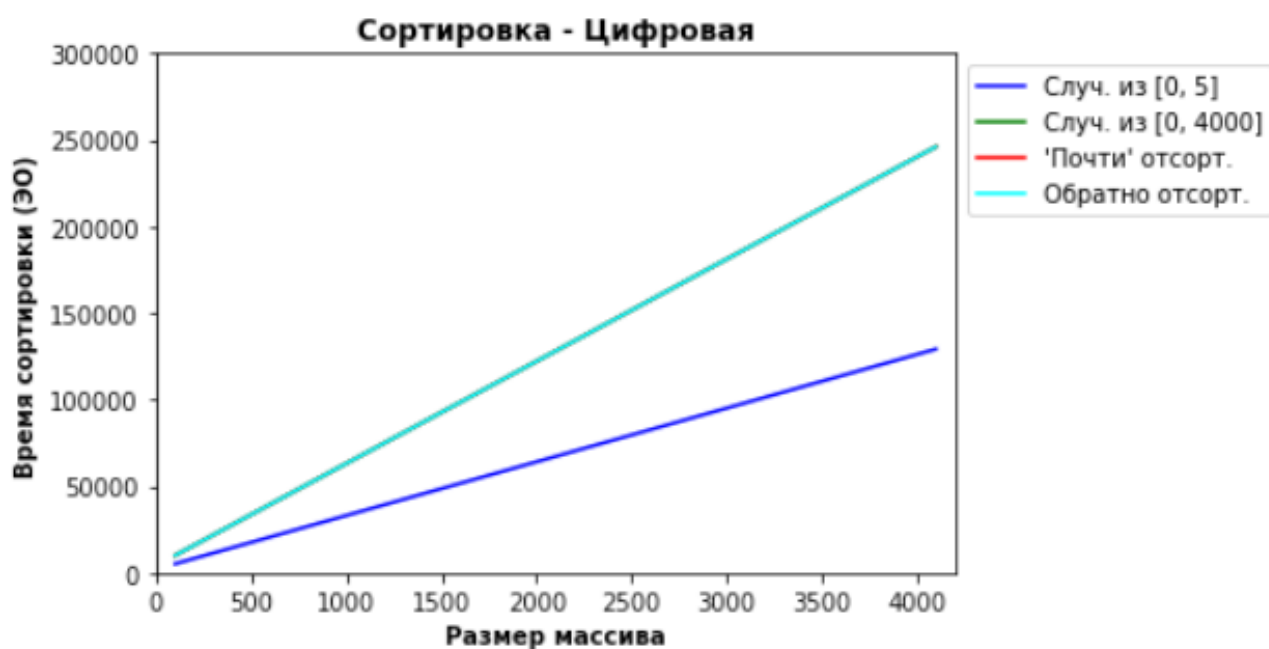


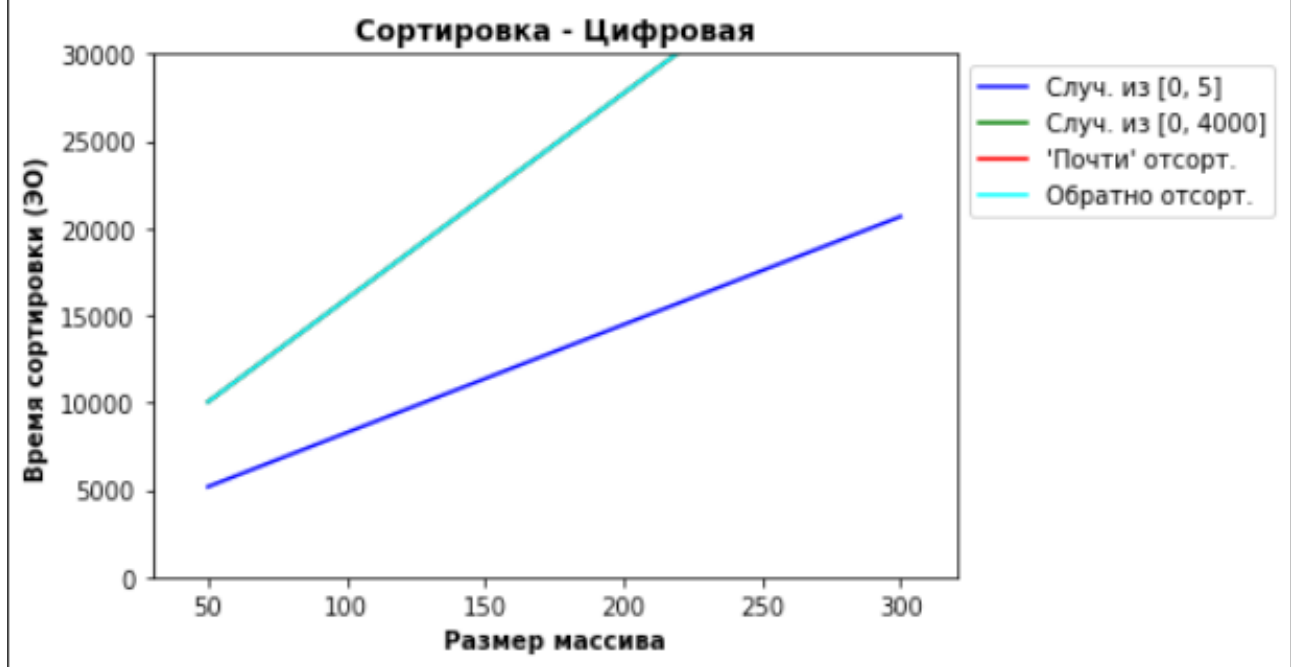
- Сортировка вставками работает намного лучше других сортировок (даже с асимптотикой $O(N \cdot \log N)$) на «почти» отсортированных массивах (неважно какой длины):



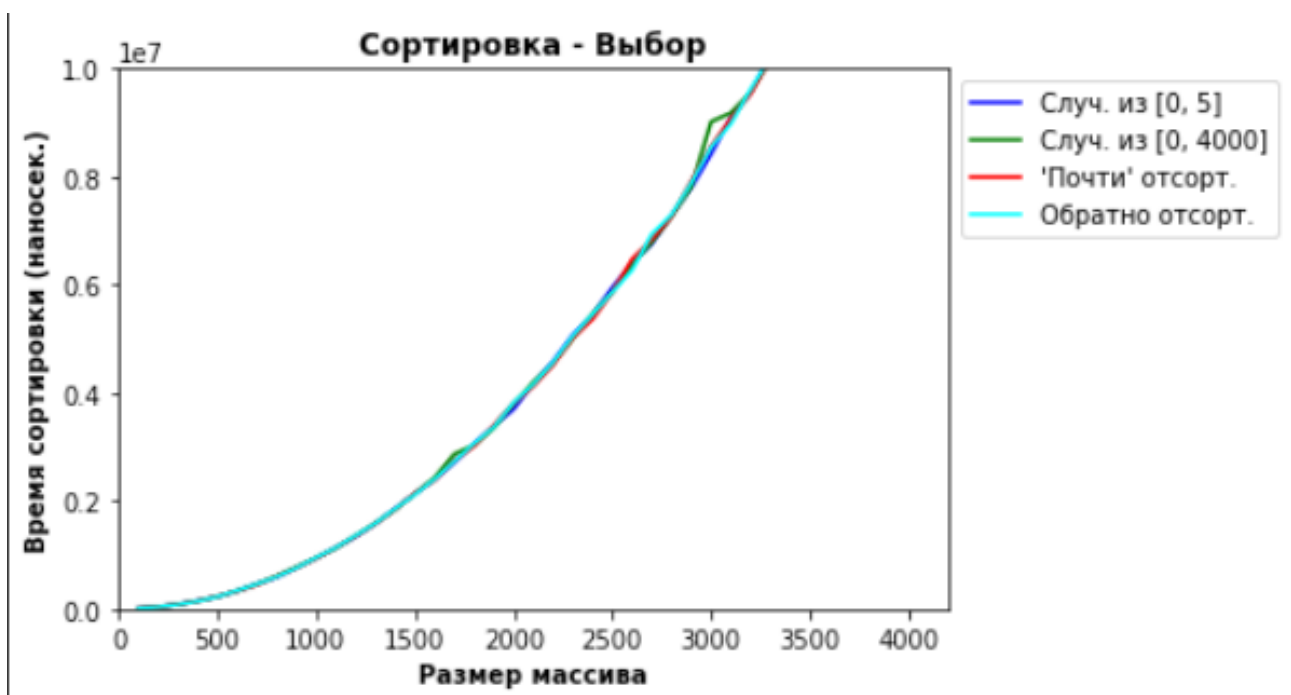


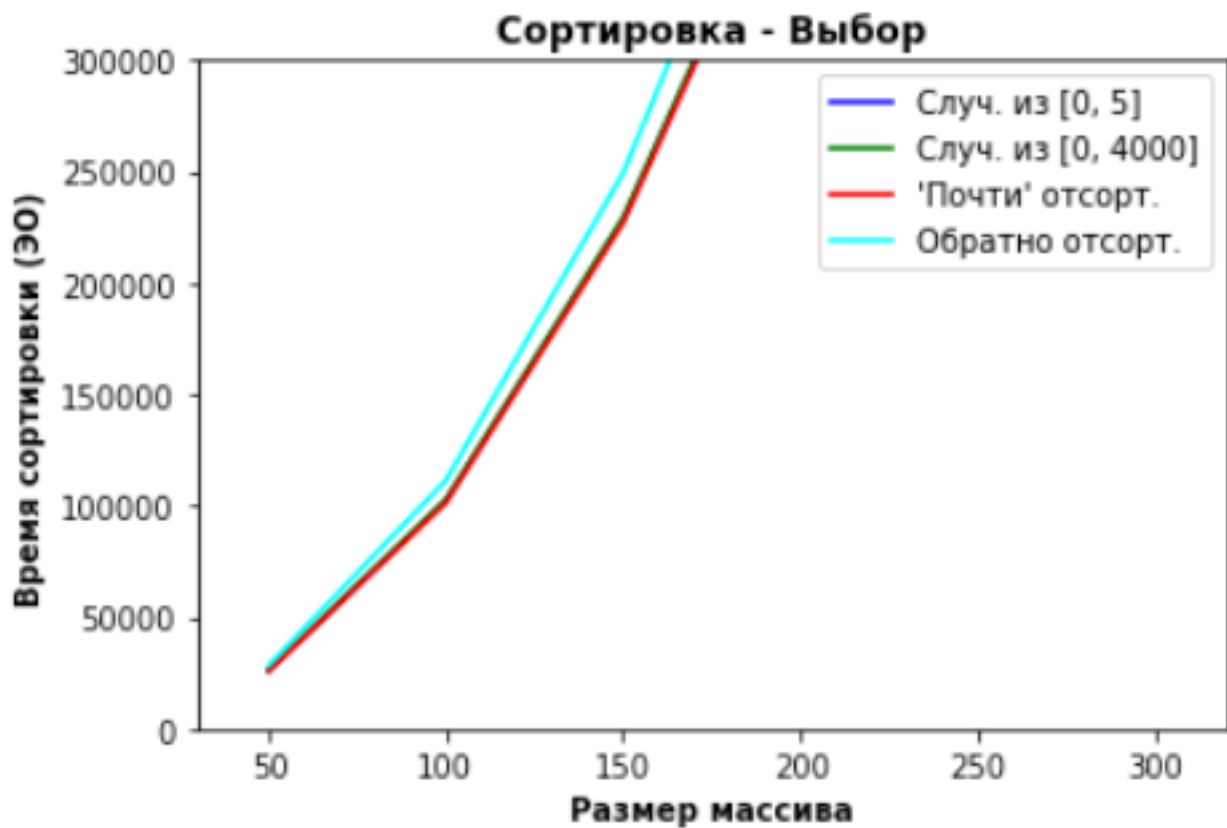
- Сортировка Шелла с последовательностью Шелла работает очень эффективно на последовательностях до 4100 элементов (научный факт, в данной работе он подтверждается графиками) (на приведенных выше графиках корректность этого факта видна)
- Цифровая сортировка работает тем лучше, чем количество уникальных значений в массиве меньше



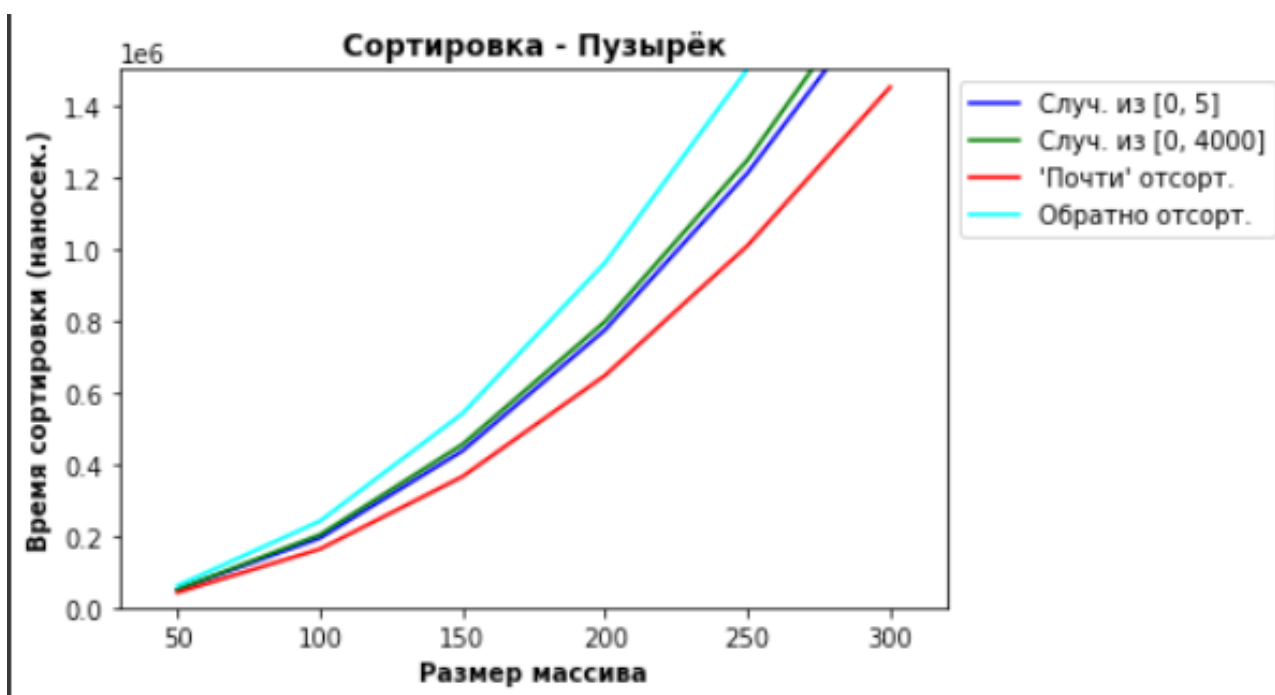


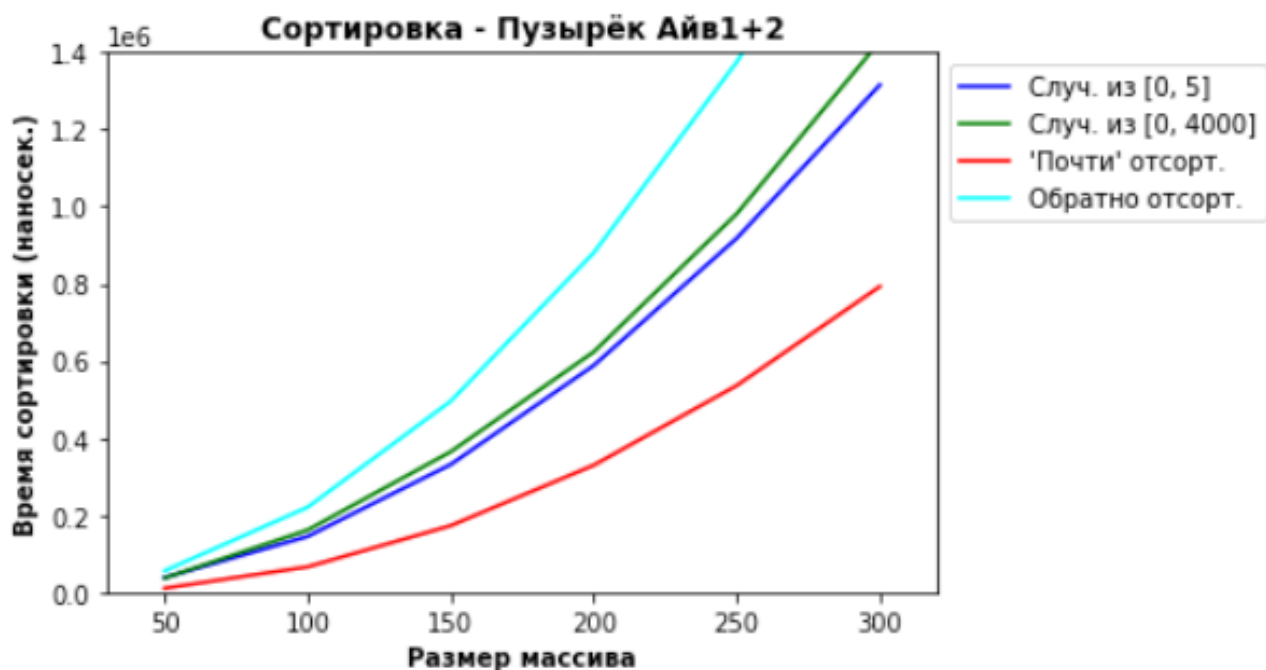
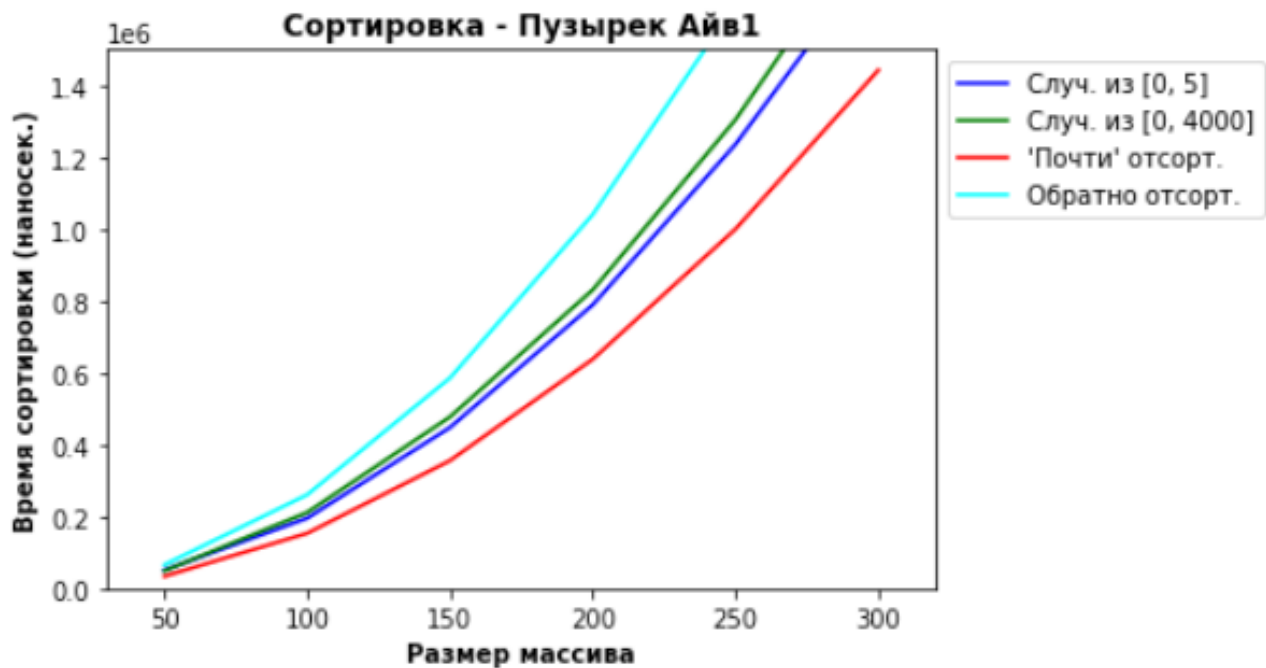
- Сортировка выбором имеет квадратичную сложность и в худшем, и в лучшем, и в среднем случаях: опорного элемента был выбран первый элемент и поэтому при делении массива на две части ("больше опорного" и "меньше опорного") будет крайне неравномерная распределённость, что и вынудит асимптотику сортировки вырождаться в квадрат.



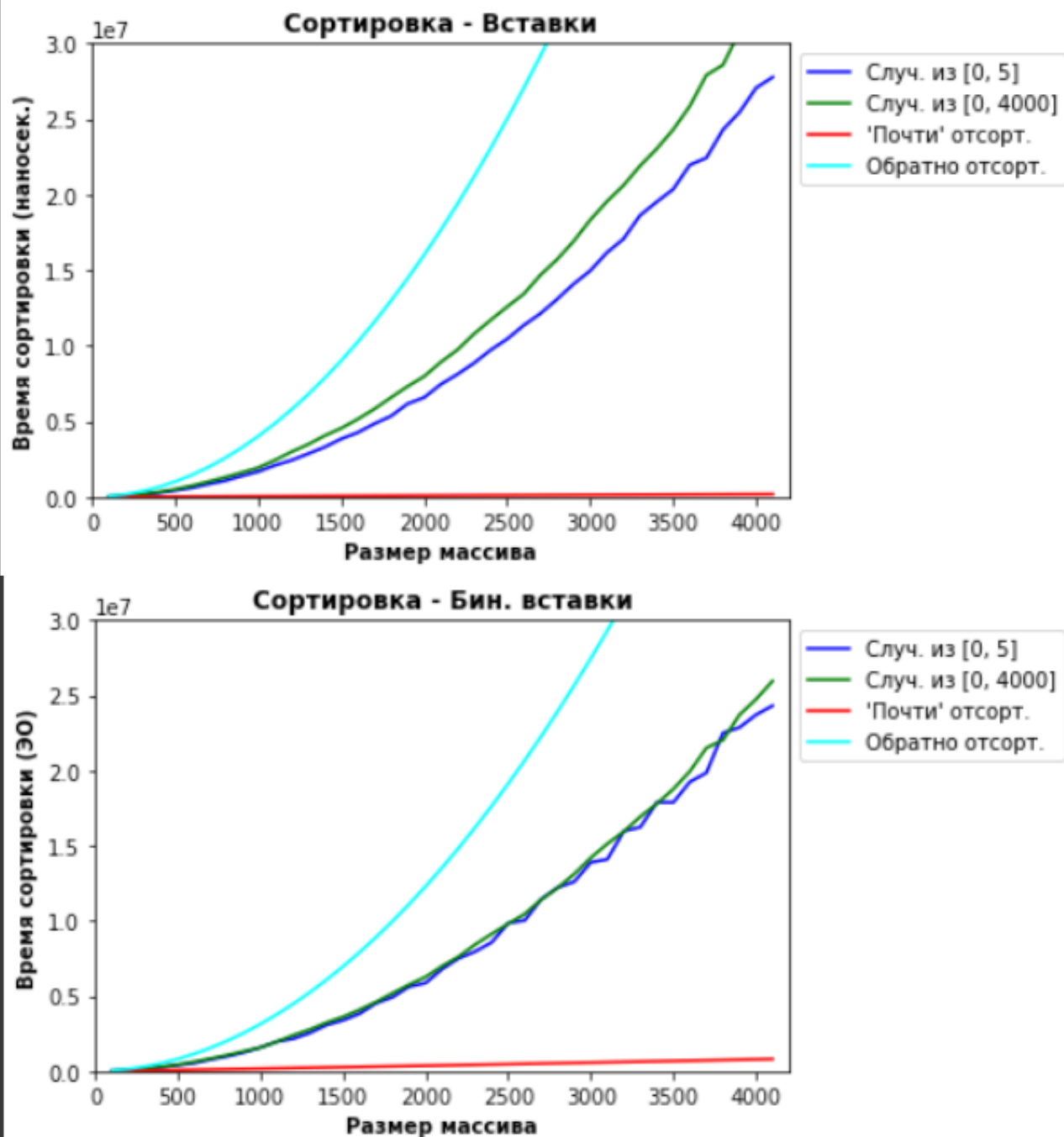


- На практике (по крайней мере, на массивах с количеством элементов от 50 до 4100) сортировка Пузырьком с условием Айверсона 1 практически не дает выигрыша по времени в отличие от условия Айверсона 2, которое прилично сокращает время сортировки. Последнее объясняется тем, что теперь мы теперь запоминаем, на какой позиции был обмен на предыдущем шаге, тем самым уменьшая число итераций на следующем шаге. Это особенно хорошо заметно на примере «почти» отсортированного массива, так как там редко происходят обмены «наиболее левых и наиболее правых» элементов.





- Бинарные вставки немного сокращают число сортировки массива, так как место для вставки элемента ищется за логарифм в отличие от линии в обычных вставках.



- Сортировка Шелла с последовательностью Циура значительно быстрее работает на «почти» отсортированном массиве, что объясняется тем, что последовательность Цифра является оптимальной для данного типа массива (максимально минимизирует количество обменов)

