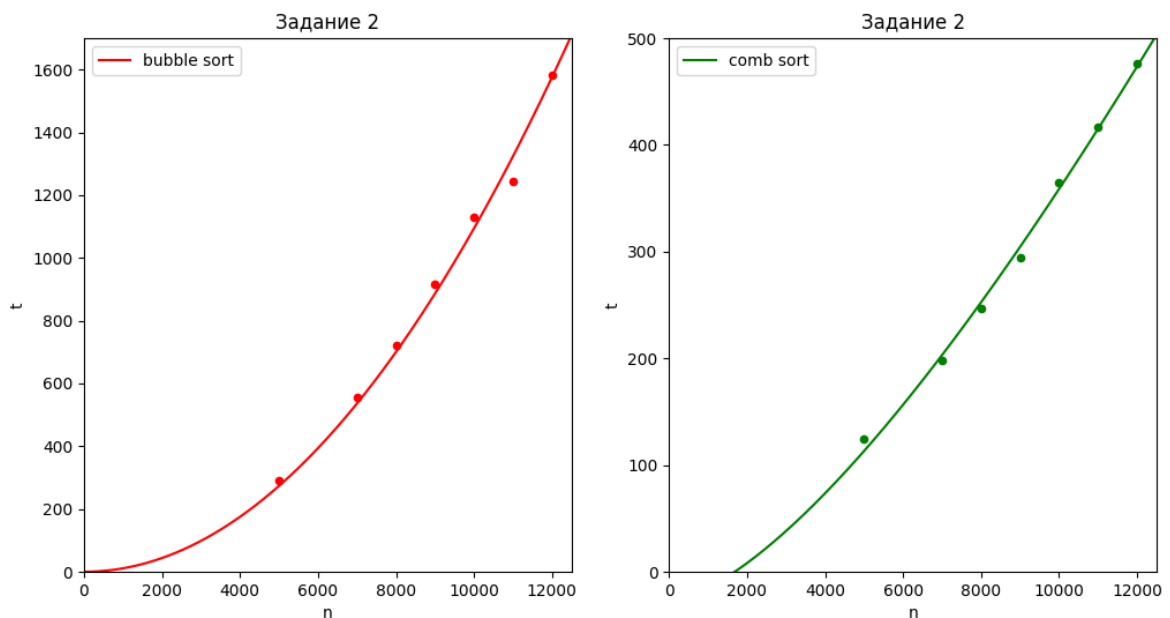
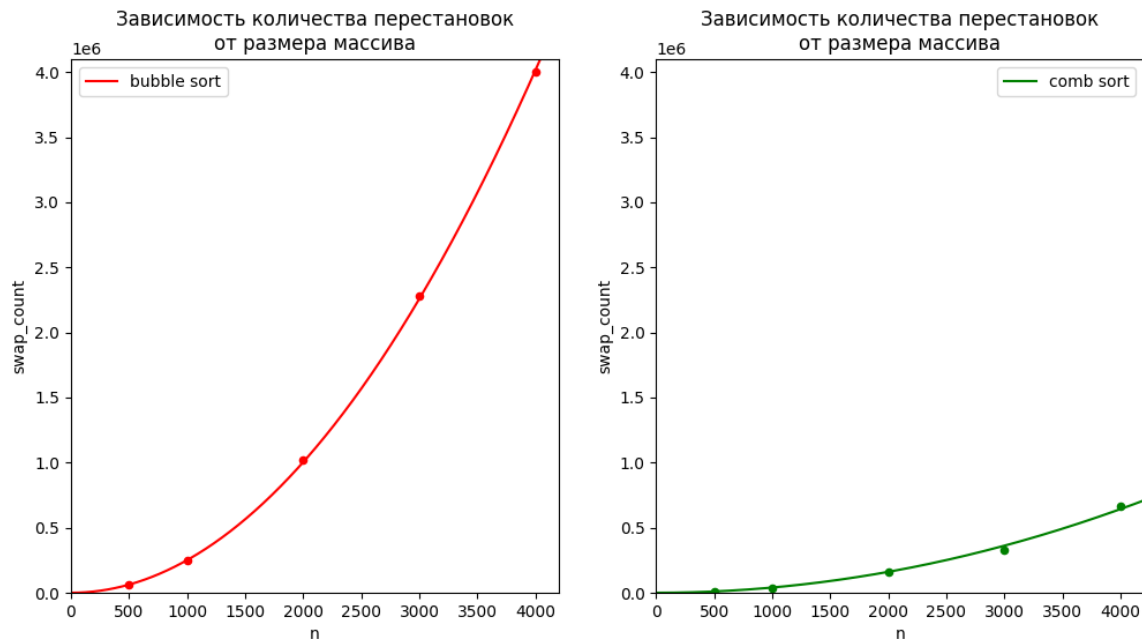


На графике зависимости времени от размера массива видно, что шейкерная сортировка работает с такой же асимптотической сложностью, как сортировка пузырьком – $O(n^2)$. Но алгоритм работает быстрее и каждый раз время шейкерной сортировки оказывается меньше.



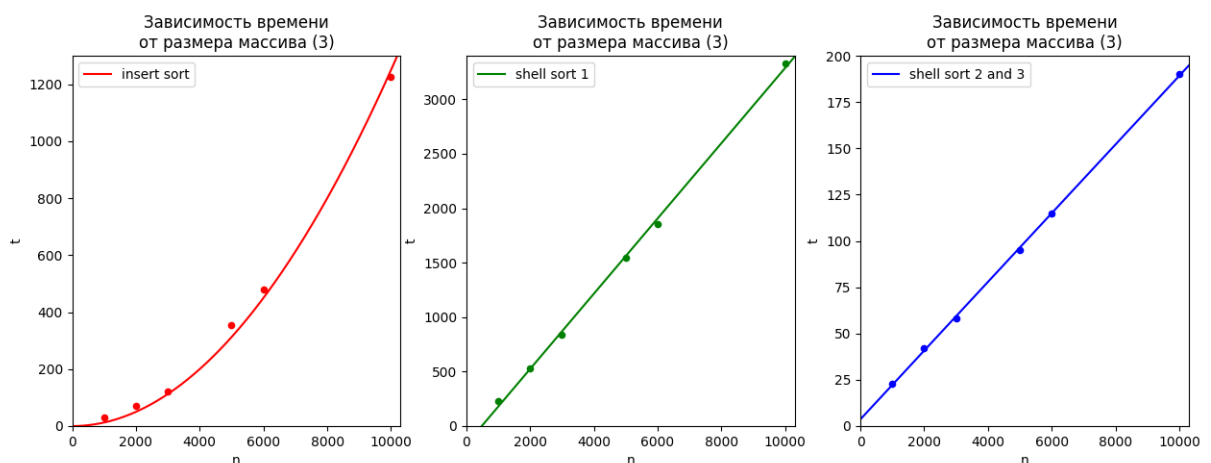
Асимптотическая сложность сортировки расческой – $O(n \log(n))$, что лучше сложности сортировки пузырьком. К тому же, улучшенный алгоритм каждый раз показывает меньшее

время, чем сортировка пузырьком.



Количество перестановок растет с увеличением размера массива как $O(n^2)$, но при сортировке расческой количество перестановок каждый раз меньше, чем при сортировке пузырьком.

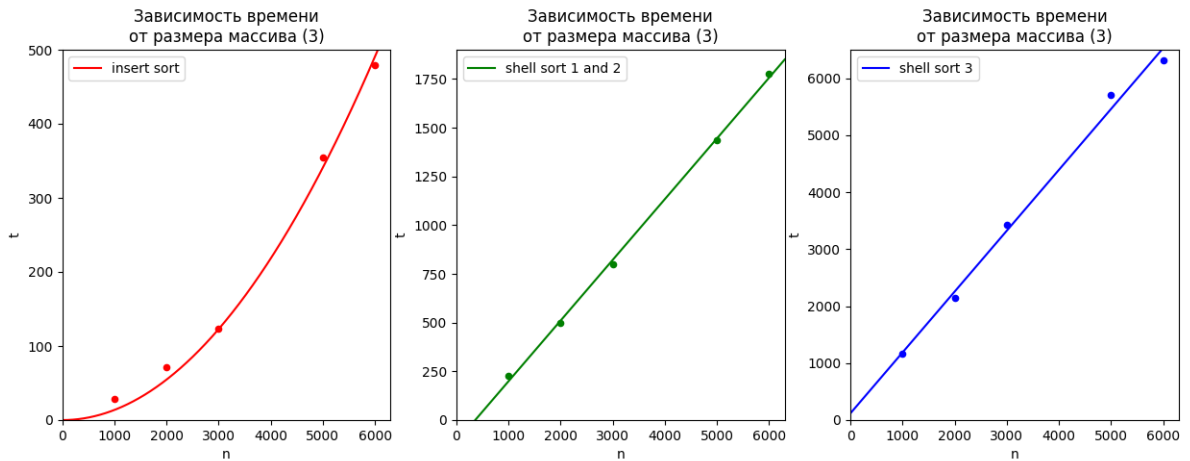
Если во втором и третьем вариантах сортировки Шелла на малых объемах данных сначала найти первый шаг с помощью библиотеки `smath` и чисел Фибоначчи, а затем запускать таймер, то результаты получаются намного лучше. Вот график, на котором использован такой подход:



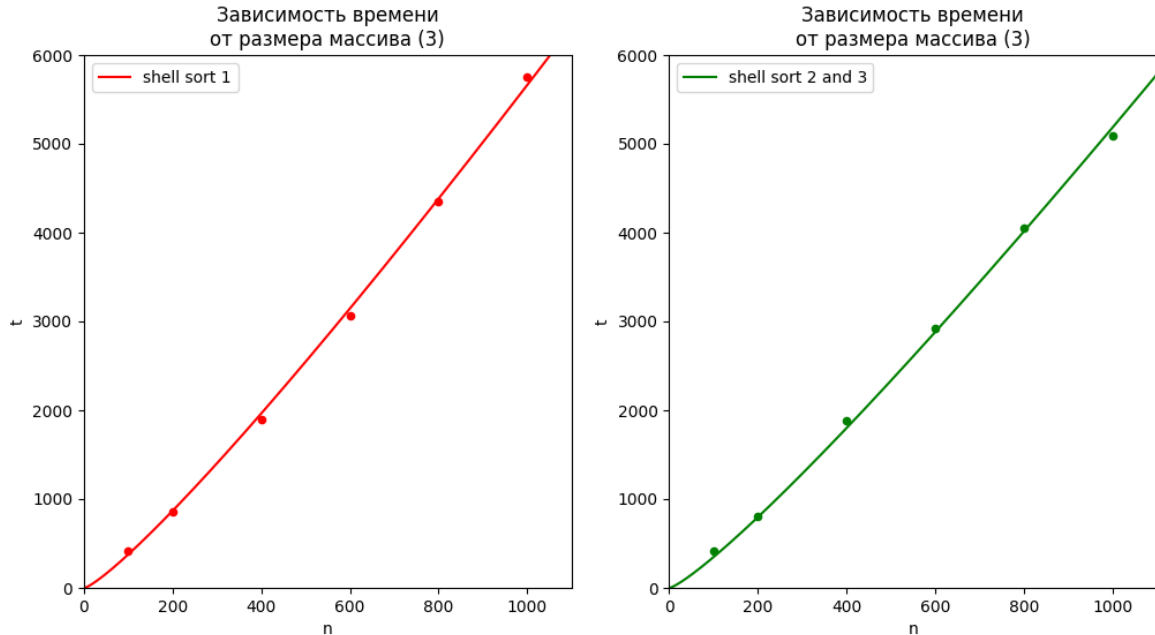
Асимптотическая сложность сортировки вставками – $O(n^2)$. Сложность всех сортировок Шелла при малых объемах данных – $O(n)$, но первый вариант работает намного медленнее второго и третьего. На малых объемах данных он работает даже хуже обычной

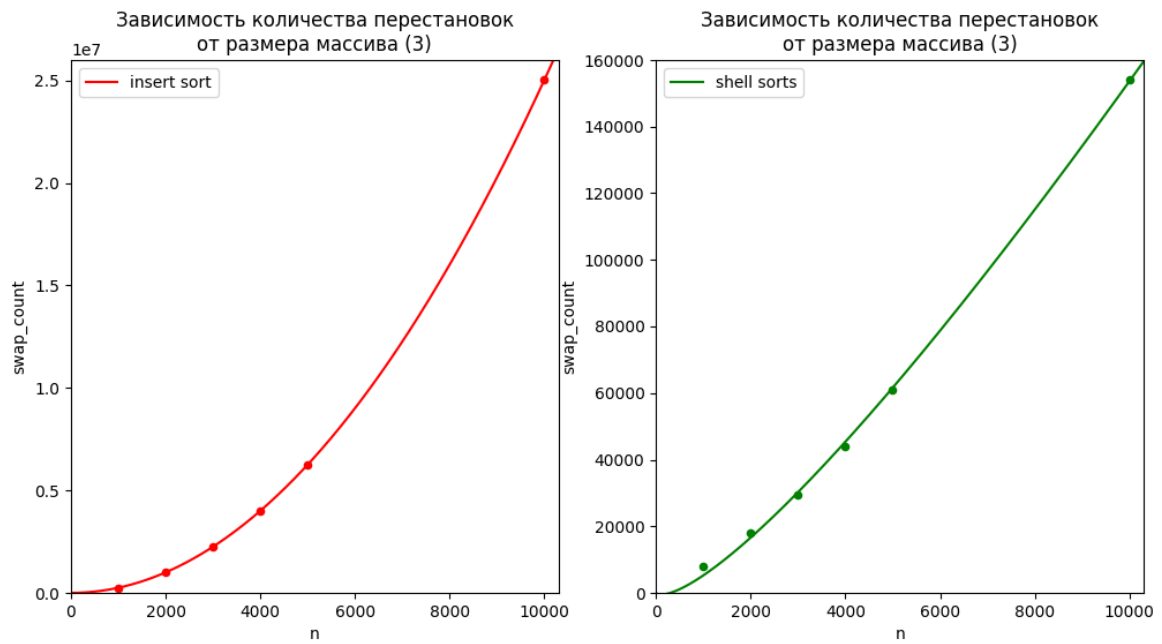
сортировки вставками, но на больших объемах за счет асимптотики он становится намного эффективнее.

Если же поиск первого шага будет внесен в таймер, то первый и второй варианты сравняются по сложности, а третий будет намного медленнее, однако сложность по-прежнему $O(n)$:



На больших объемах данных (более 1 млн элементов в массиве) асимптотическая сложность меняется и становится равной $O(n \log(n))$:





Количество перестановок при сортировке вставками растет с увеличением размера массива как $O(n^2)$, а при любом варианте сортировки Шелла как $O(n \log(n))$. К тому же, при сортировке Шелла количество перестановок каждый раз меньше, чем при сортировке вставками.

Итак, если вынести поиск первого шага за пределы таймера, то первый вариант сортировки Шелла по времени хуже второго и третьего, которые примерно одинаковы. По количеству перестановок все три сортировки примерно равны.