

# **Задание А2. Гибридная сортировка MERGE+INSERTION SORT**

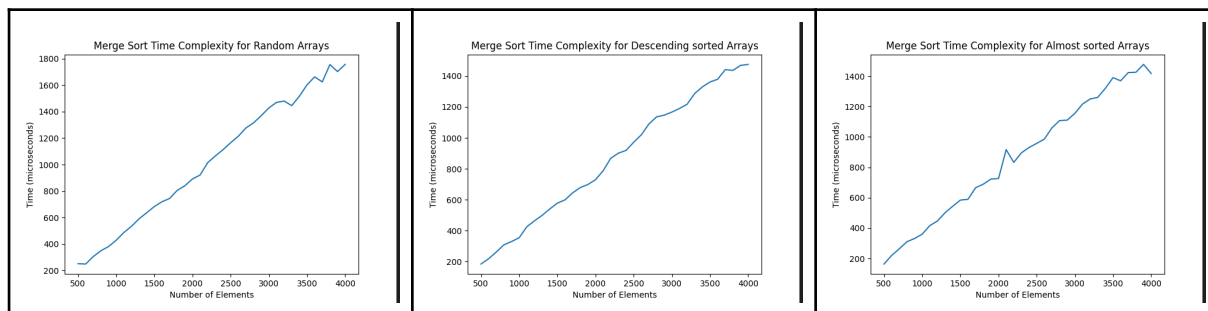
**Мирошниченко Денис БПИ225**

**ОС: Ubuntu (linux)**

**CPU: i5 12450H**

**RAM: 16 GB**

## MERGE SORT:

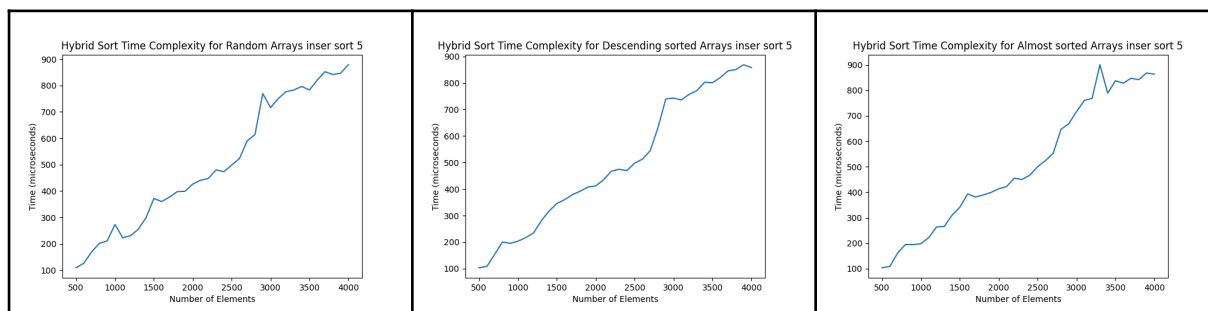


Как мы видим все графики получились очень похожие, возможно, потому что максимальное число элементов в массиве 4000 (это совсем ничего для компьютера).

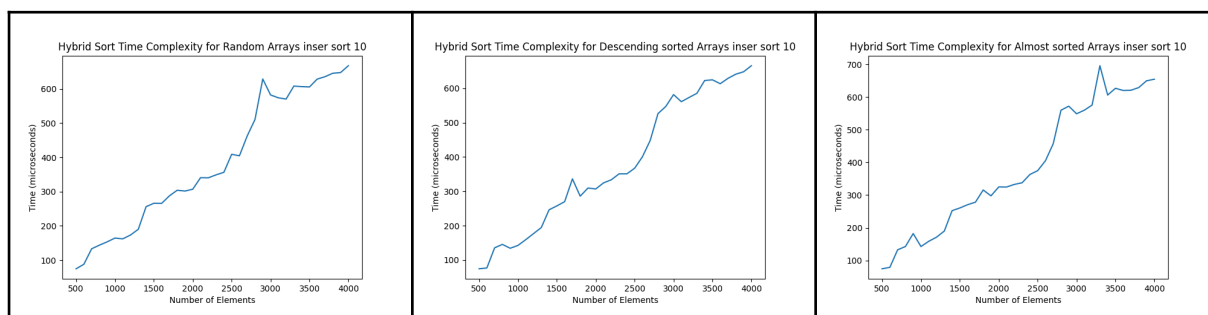
Быстрее всего отработало на массиве, который отсортирован по убыванию (200-1400 микросекунд), далее по скорости идет, если массив почти отсортирован (200-1400 микросекунд), но по сравнению с отсортированным по убыванию при стремлении числа элементов к бесконечности, время начинает увеличиваться быстрее, чем на отсортированном по убыванию. Самый медленный оказался массив с случайными значениями (300 - 1800 микросекунд).

# Hybrid sort

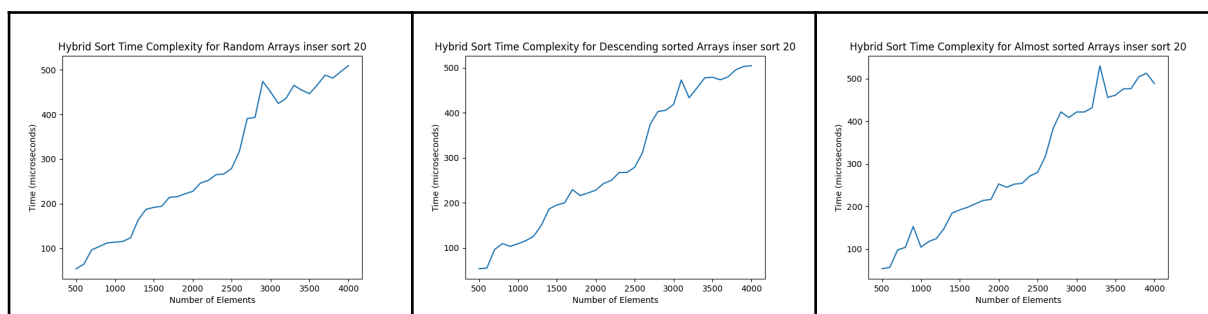
Insertion sort начинается с 5 элементов:



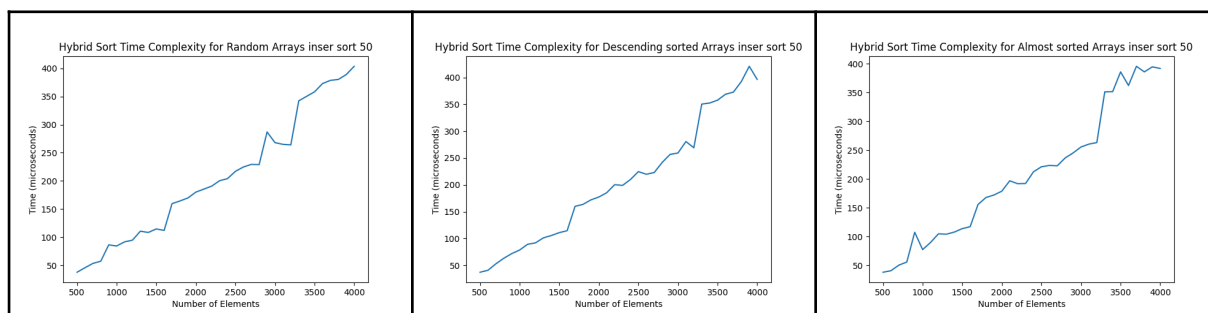
Insertion sort начинается с 10 элементов:



Insertion sort начинается с 20 элементов:



Insertion sort начинается с 50 элементов:



Как мы видим из графиков для каждого момента insertion sort (5 10 20 50) они разные, самая минимальное время у insertion sort 50. Также по данным видно, что HybridSort работает быстрее: где-то на 1.5 - 2 раза (это очень значительный прирост). Еще хочется выделить, что на любых деления для insertion sort, hybrid работает совсем по-разному (по времени), например, для данных размером 4000 гибридная сортировка с параметром insertion sort равным 5 показывает результаты приблизительно 800 микросекунд, а с 50 результат равен около 400-450.

Вывод: Hybrid эффективнее, чем обычный Merge, но по моему мнению можно найти более оптимальную границу для insertion (не среди представленных, возможно, больше, чем 50), чтобы этот алгоритм стал еще более эффективным.