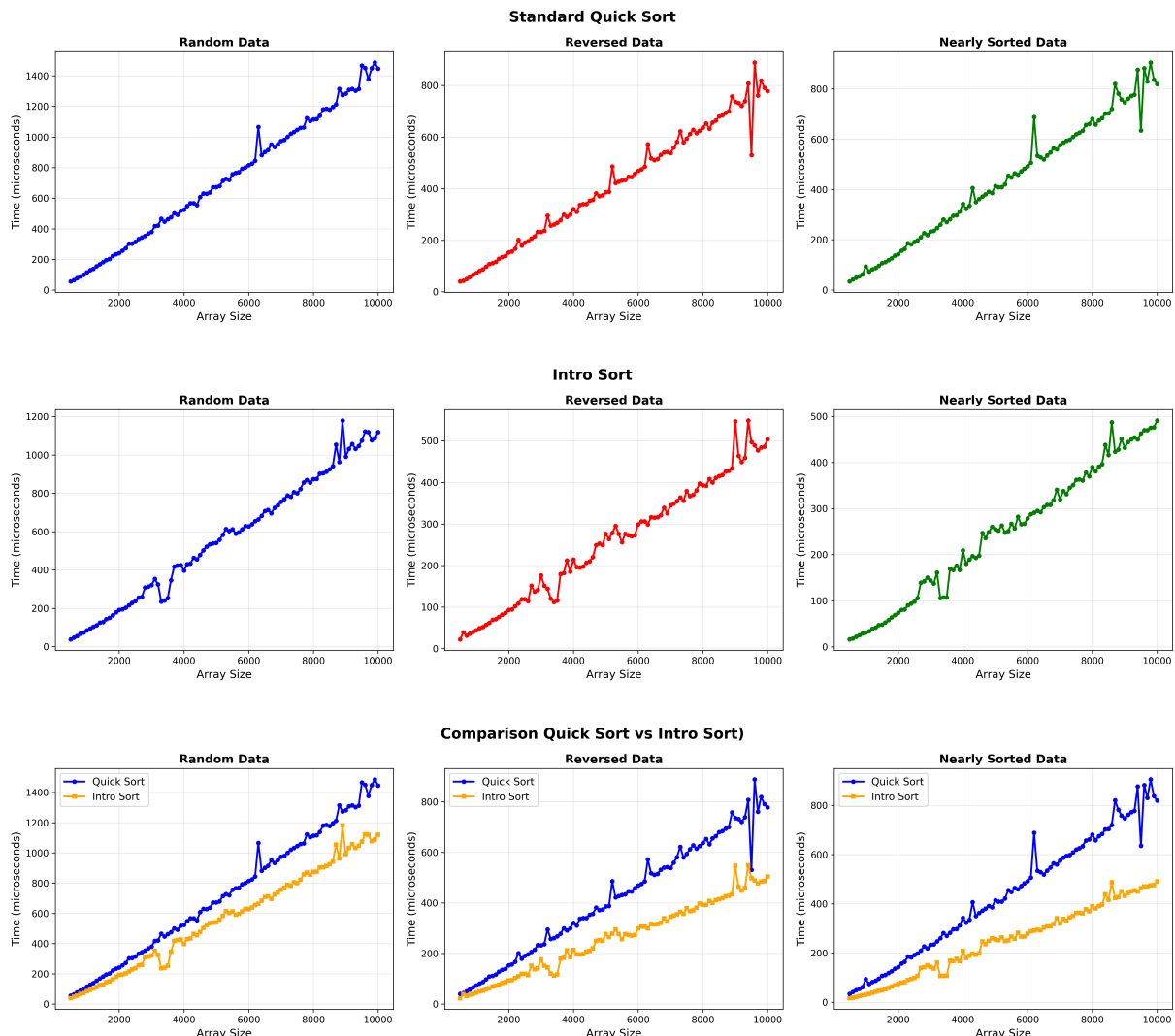


Solution A3

ID: 349331258 [Репозиторий гитхаб](#)



По графику сравнений двух алгоритмов, видно, что чем больше становится размер массива, тем больше видна разница между Quick Sort и Intro Sort.

Intro действительно оказывается быстрее обычно Quick, благодаря тому, что при достаточно маленьких n (< 16), мы вызываем обычный Insertion sort, уменьшая рекурсивные вызовы. HeapSort реализована итеративно, следовательно, когда глубина рекурсии превышает $2 \cdot \log_2(n)$, мы начинаем использовать сортировку кучей. Что также уменьшает стек рекурсивных вызовов.

На случайных массивах разница не очень большая, а вот в случае обратно отсортированных массивов можно увидеть заметную разницу. Происходит это из-за того, что обычный Quick Sort может выбирать плохой pivot, примерно в конце, тогда худший случай это $O(n^2)$. В Intro Sort же мы быстро достигнем $2 \cdot \log_2(n)$, а дальше будем сортировать с помощью Heap Sort, благодаря чему получаем асимптотику $O(n \cdot \log(n))$.

На массивах почти отсортированных, мы будем выигрывать благодаря Insertion sort, так как на почти отсортированных массивах оно будет отрабатывать за $O(n)$

По итогу, можно сказать, что Intro Sort будет быстрее Quick Sort. Таким образом мы можем избавиться от худшей сложности, превратив ее из $O(n^2)$ в $O(n \cdot \log(n))$.