**Teamwork**

1. The worst case of quick sort.

The worst time complexity of quick sort is n2 and it happens when after the pivot is chose, all elements that need to be picked are in the same side of the pivot, so each time there are only one element is sorted and this method needs to pick up for pivot for n time (n is the number of the element) and every time all unsorted elements need to compare with the pivot that would cause n time as well. Therefore, the total worst-case big-Oh complexity is n square.

The situations that causing the worst-case scenario are most due to the unlucky (or unwisely) choosing the pivot. To solve this, the paper gives us a very insightful way, choosing the pivot as the medium one of three elements of the unsorted list.

Another thing that I found that is very interesting in the paper is that even if the big-Oh complexity is the same, the actual running time of different types of sorting algorithms is different. For quick sort and merged sort, as quick sort swaps less than merge sort, it runs faster. This also happened among insertion sort, bubble sort and selection sort.