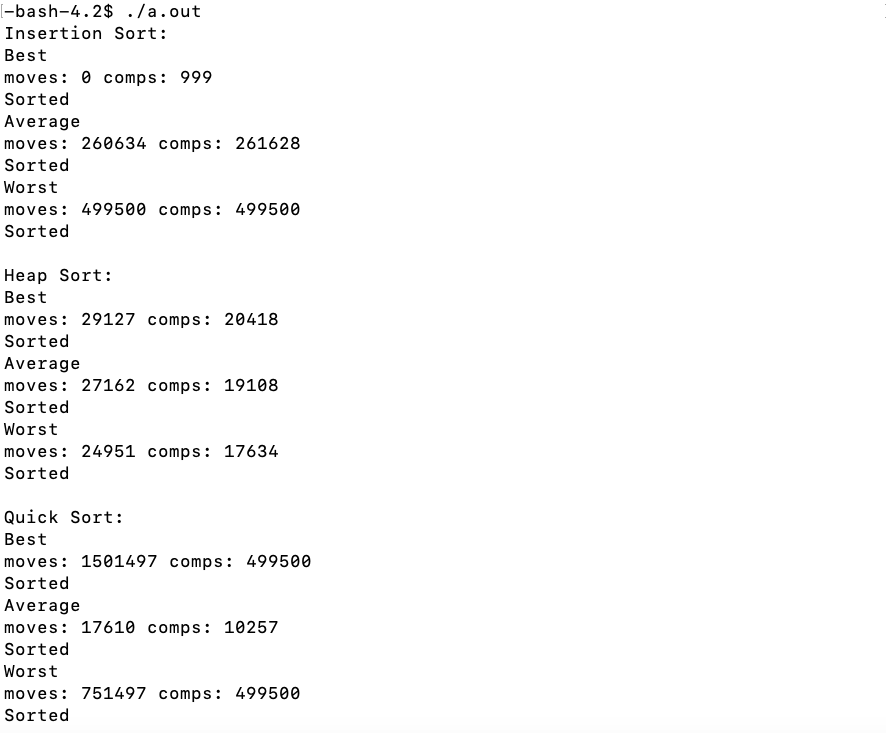
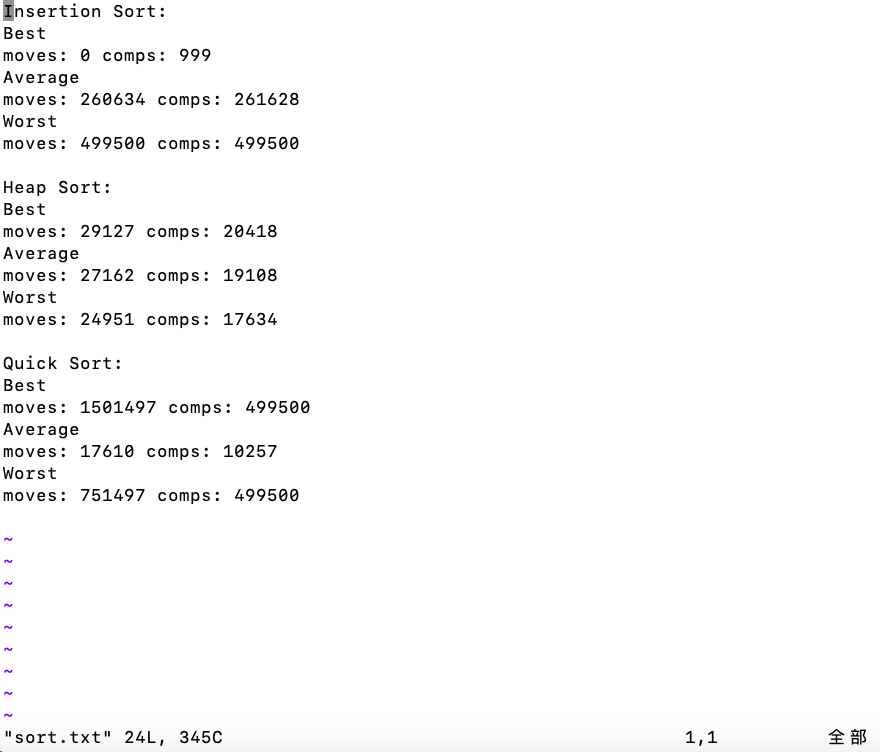
Result in COE Linux:

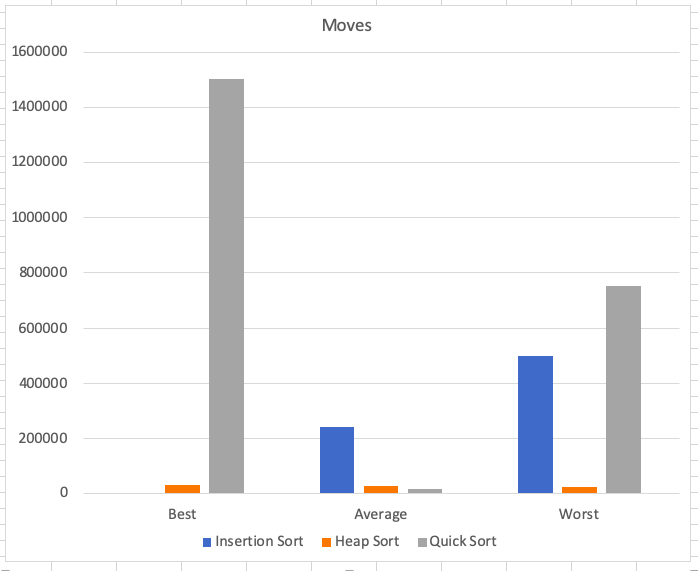


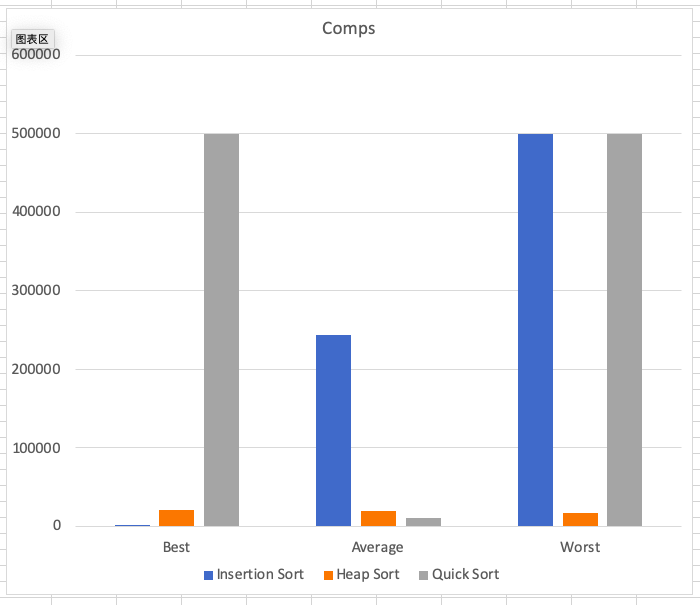
Result in sort.txt



Answer:

1. Because Insertion Sort moves small integer to the front. And when the array is already been sorted, no integer needs to be moved. So moves = 0.
2. In insertion sort, there are 999 circles to move integers to right situation. When the array is sorted, each circle ends with one comparison. So the comparison times in best situation is 999.
3. Here are the excel graphs.





In best case, insertion sort performs best and quick sort performs worst in both moves and comparisons.

In average case, quick sort performs best and heap sort performs well, and insertion sort performs very bad in both moves and comparisons.

In worst case, heap sort performs best. Quick sort and insertion sort performs bad, especially in comparison times.

In most of times, the arrays we get are always average or worst case, so heap sort and quick sort should be chosen, and we could choose suitable algorism according to the characteristic of array.