Report

Quick sort is a much more efficient algorithm for sorting large data than insertion sort.But, insertion sort is faster for small data because Quick Sort has extra overhead from the recursive function calls. Insertion sort is also more stable than Quick sort and requires less memory. Based on this information, I wrote a new sort. I divide the data into smaller data using the quick sort algorithm and sort these data with insertion sort. I named this algorithm is hybirdSort.
I chose the threshold point as 9 for array. Because the threshold point for when the running time is significantly better in insertion sort than that of quicksort is around 9.

Analysis ;

In the best case; If the incoming data size is less than 9, which is the threshold point, and is in order.Then only insertion sort execute and time complexity of this situation is;

Best case = **O(n).**

In the avarage case; When the number of elements is less than some threshold k(9),then we stop. This process takes Θ(nlogn) time in total for the quick sort part.Later when the whole array has been processed, each element will be at most k positions away from its final sorted position.Now if we perform insertion sort on it, it will take O(kn) time to finish the sort, which is linear as k is a constant.Then time complexity of this situation is;

Avarage case = Θ(nlogn+n) = **O(nlogn)**

In the worst case; the worst case would occur when the array is already sorted in decreasing or increasing order. When the number of elements is less than some threshold k(9),then we stop. This process takes $O(n^2)$ time in total for the quick sort part.Then perform insertion sort on it, it will take O(kn) time (proved in the average case description).

Then time cimplexity of this situation is ; Worst case = $O(n^2+n)$ = **$O(n^2)$**