CSCI 58000 Program assignment 2
Zhihao Cao
Nov. 8 2016

Dimemsions	r = 4000	s = 25
General sort used	**Quick sort**	
	General sort time	Column sort time
Trial 1	0.042s	5.922s
Trial 2	0.043s	5.890s
Trial 3	0.046s	5.909s
General sort used	**Insertion sort**	
Trial 4	10.531s	1.260s
Trial 5	10.533s	1.262s
Trial 6	10.526s	1.268s

By reviewing the trial 1-3 results, the Column sort using quick sort is not only less efficient, but also cost more memory spaces than general quick sort.
The reason for causing this unexpected result is that the general quick sort runs on $O(n^2)$ when the input is nearly sorted (worst-case). The procedures of the column sort using quick sort applies sorting operations 4 times which three of them are sorted-like input. That is why the performance is lower than general quick sort.
Therefore, Column sort using quick sort is not a good choice.

By reviewing the trial 4-6 results, the Column sort using insertion sort is more efficient than general insertion sort, but costs more memory spaces.
The reason why the Column sort using insertion sort is faster than general insertion sort is that the insertion sort runs on $O(n)$ when the input is nearly sorted (best-case). As explained above, the Column sort applies sorting operations 4 times which three of them are sorted-like input. That is why the performance is higher than general insertion sort.
Therefore, Column sort using insertion sort is a considerable choice when insertion sort is required and memory space is not concerned.