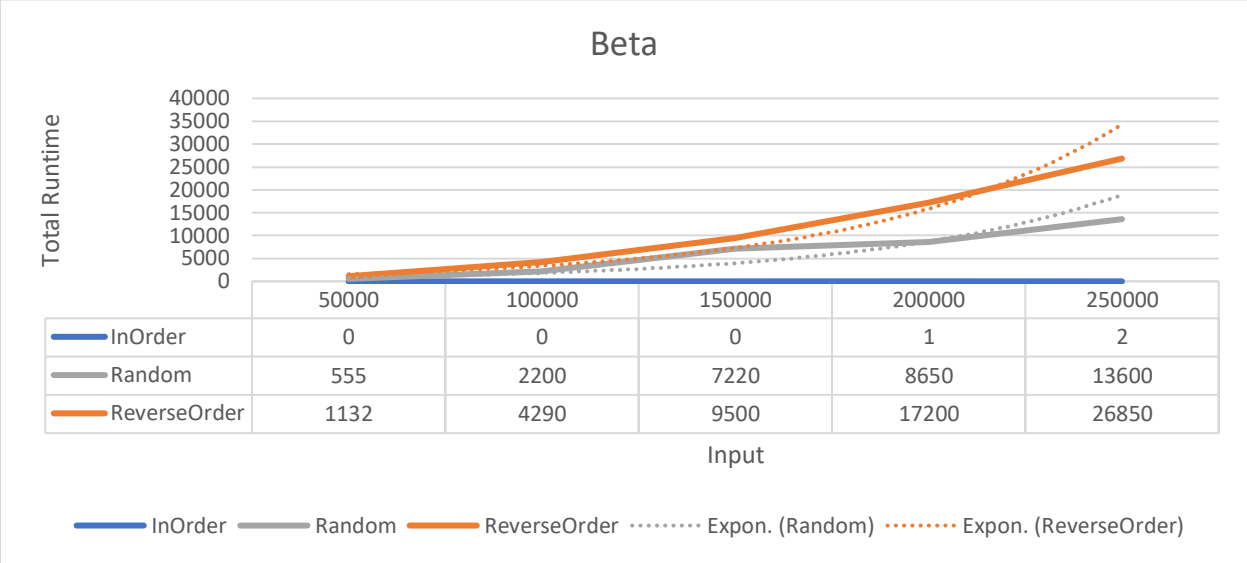
Kaiyin Chen

CS241

October 30, 2020

HW4 Task 2 – Sort Detective

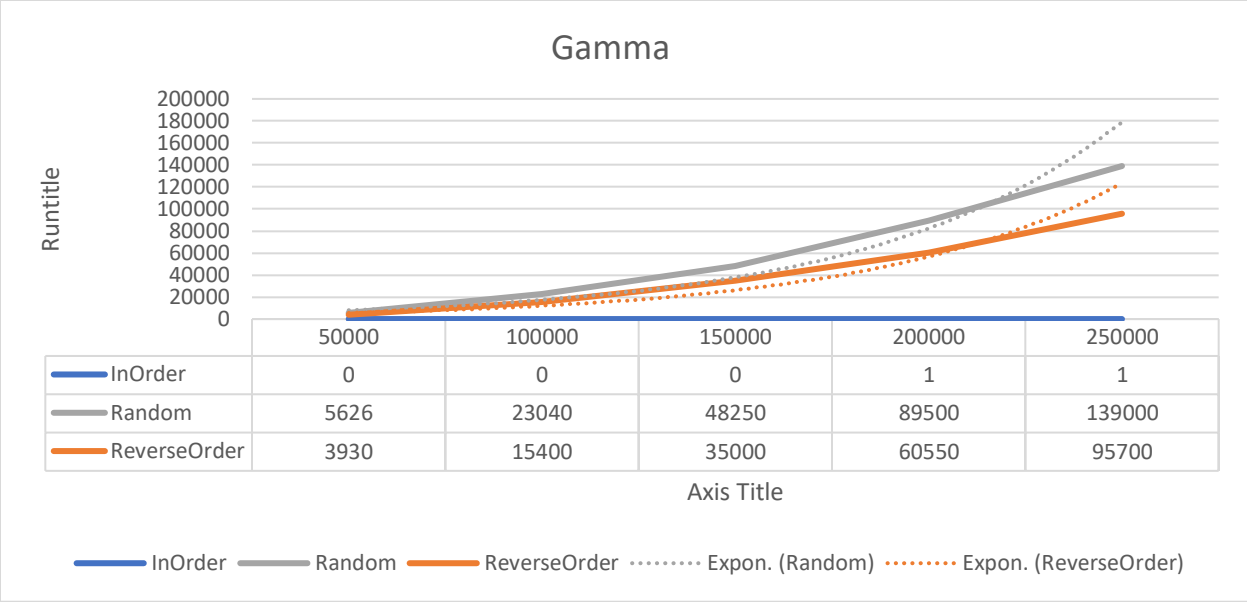
Which Sort Is Which?



For the InOrder sort of Beta method, the total runtime increasing very slowly and mostly constant as the inputs are increasing. Therefore, the Big-O for InOrder data type in Beta sort is O(N).

For the Random and ReverseOrder data types on Beta sort, the runtimes are increasing as the inputs increasing. The growth rate for random date type is 61.7% and for ReverseOrder data type is 47.6%. In conclusion, the runtime for Random and ReverseOrder data types are increasing in exponential time and the Big-O for those data types is $O(N^2)$.

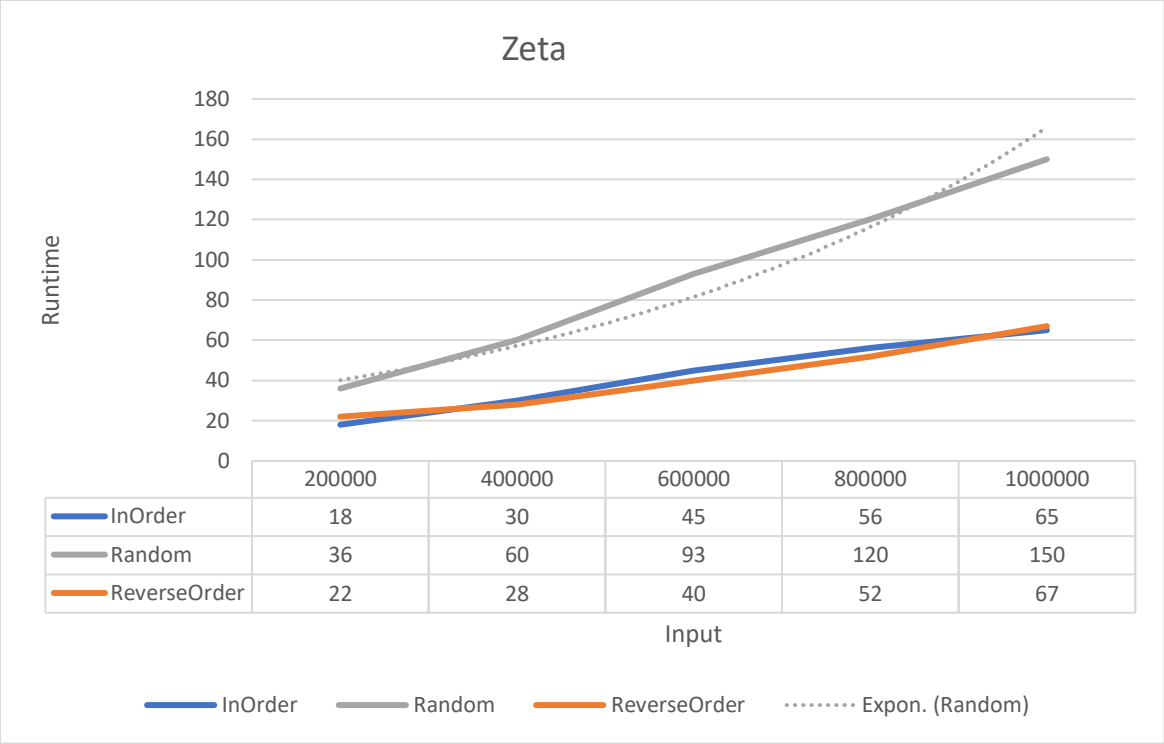
Insertion sort and bubble sort have the same big-O, but the major different between these two sorts are the real speed they ran. Insertion sort is faster than bubble sort because the number of swaps less than bubble sort. Therefore, Beta is an insertion sort.



In InOrder data type of Beta sort, the runtime is increasing very slowly and mostly constant. Which the Big-O is O(N).

In Random and ReverseOrder data types, the runtime is increasing as the input increasing. In both data types, they are growing in exponential time and the growth rate is 73%. The Big-O of both the Random and ReverseOrder data types is $O(N^2)$.

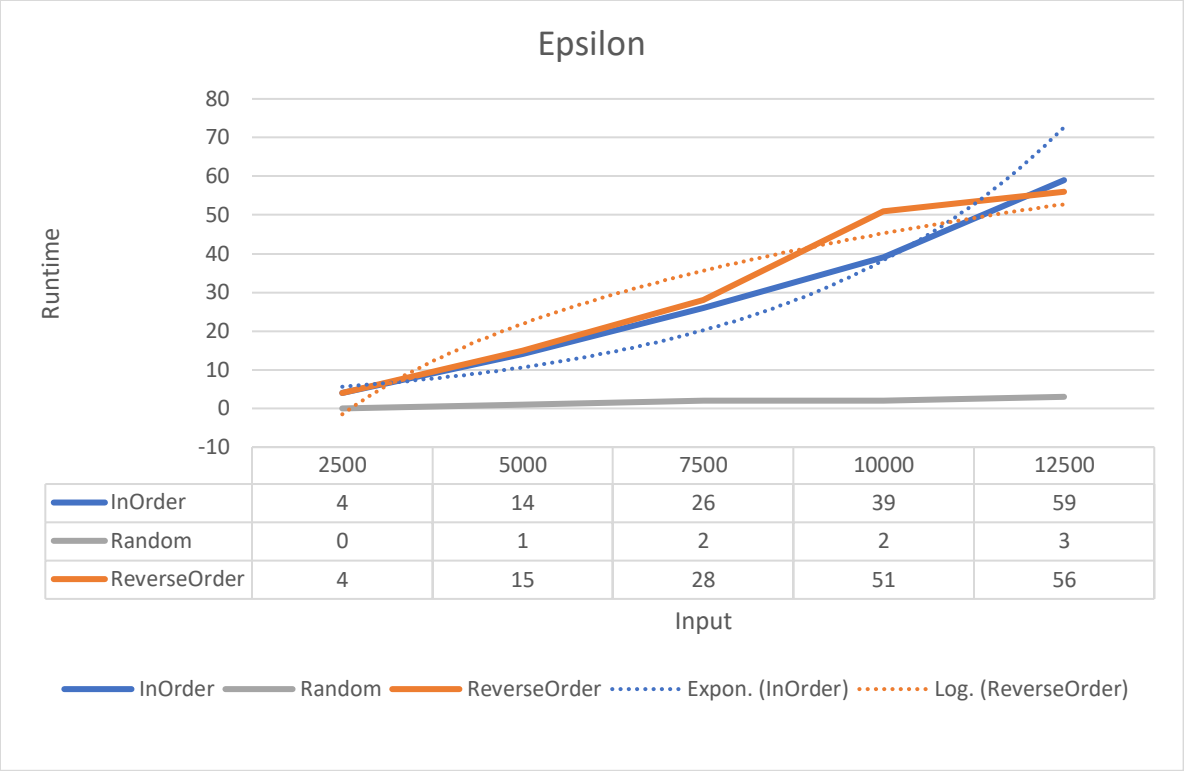
Gamma is a bubble sort instant of insertion sort because bubble sort is slower than insertion sort. Because bubble sort is comparing and swapping elements in every pass. Compared the Beta table and Gamma table, all the runtimes in Gamma are faster than the runtimes in Beta. Therefore, Gamma is a bubble sort.



For Random data type in Zeta sort, the runtimes are increasing as the inputs increasing. The growth rate is 30% and increasing in exponential time. The Big-O is $O(N^2)$. For InOrder and ReverseOrder data types in Zeta sort, the runtimes are increasing as the inputs increasing. The runtimes for these two data types are similar in the same inputs and are grow in linear time, the Big-O is O(N). Zeta is a unknow sort.

For InOrder, Random, and ReverseOrder data types in Theta sort, the runtimes are increasing as the inputs increasing. The Random data type are run slower than InOrder and ReverseOrder, and the runtimes are mostly the same in InOrder and ReverseOrder data types. Therefore, the Big-O for InOrder, Random, and ReverseOrder data types are O(NLogN) and they are increasing in logarithmic time.

Merge sort is running in consistent speed in any size of data, this mean in large size data it will more efficiency. Its preferred working in Linked Lists and is a stable sort. Therefore, can conclude Theta is a merge sort.



For Random and ReverseOrder data types in Epsilon sort, the runtimes are increasing as the inputs increasing. The growth rates in Random data type are decreasing from 200% to 40%, this means it was increasing in logarithmic time. The growth rates in ReverseOrder are from 116% to 9%, it shows it was increasing in logarithmic time. Therefore, the Big-O in Random and ReverseOrder is O(NLogN).

For InOrder data type in Epsilon, the runtimes are increasing in exponential time as the input increasing. The Big-O is $O(N^2)$.

Quick sort work faster in small data and not work efficiency in large data. It was preferred to work for Arrays and is not a stable sort. In Epsilon, large inputs that more than 20000 needs take much longer time run and sometimes it will break. Therefore, it could conclude Epsilon is quick sort.