When we sort an array whose elements are 10000,9999,9998,…,3,2,1, Sorter A,B,D,E all take time less than 1 seconds. Only Sorter F take an average time of about 40 seconds. Since using Bubble sort to sort this strictly decreasing array requires the most number of read and write operation, Bubble sort takes the most time to sort this kind of array. So F is Bubble sort.

Insertion sort

When we sort an array whose elements are 1,2,3,…,9998,9999,10000, Insertion sort takes O(n) time which is the fastest. Indeed when we run tests, using Sorter B to sort this array gives the lowest average time of below 0.0001 seconds. Whereas other sorters give average runtime of above 0.001 seconds. So Sorter B is insertion sort.

When sorting different kinds of array of the same size, merge sort gives the most relatively constant and stable runtime. When we run tests on different kinds of array of the same size, sorter E gives the most relatively constant and stable runtime. And also merge sort has the best upper bound runtime among all the sorting algorithms so, and indeed we have merge sort giving the relatively best runtime. So sorter E is merge sort.

Sorter A is faster when we increase the size of array by 10 times as compared to sorter D. runtime of sorter A increase by factor of n whereas D increase by n square,