

COSC 2P03

Assignment 3

June 20th, 2018

Sawyer Fenwick

6005011

I sorted an array of 1000, 5000, and 10000 values, repeated 10 times each, each time being randomly filled with an upper bound of 10 million and a lower bound of 0. In the chart below the left side is how long it took to sort the unsorted array and the right side is how long it took to sort the sorted array. For larger values of n insertion sort becomes ineffective compared to the other sorts (when sorting a random array not a sorted array). When sorting an already sorted array quick sort takes much longer than other array but sorts the random array faster than any of the other sorts.

n	Heap Sort $O(n \log n)$	Quick Sort $O(n^2)$	Merge Sort $O(n \log n)$	Radix Sort $O(nk)$	Insertion Sort $O(n^2)$
1000	0ms/0ms	0ms/1ms	0ms/0ms	0ms/0ms	0ms/0ms
5000	0ms/0ms	0ms/11ms	0ms/0ms	0ms/0ms	2ms/0ms
10000	1ms/0ms	0ms/58ms	1ms/0ms	1ms/1ms	8ms/0ms

Based on the chart, it seems that each of the sorting algorithms run at the recorded Big-O run time.