Uzair Akram

Algorithms

Assignment 1

Part 1 Sorting Algorithms - report

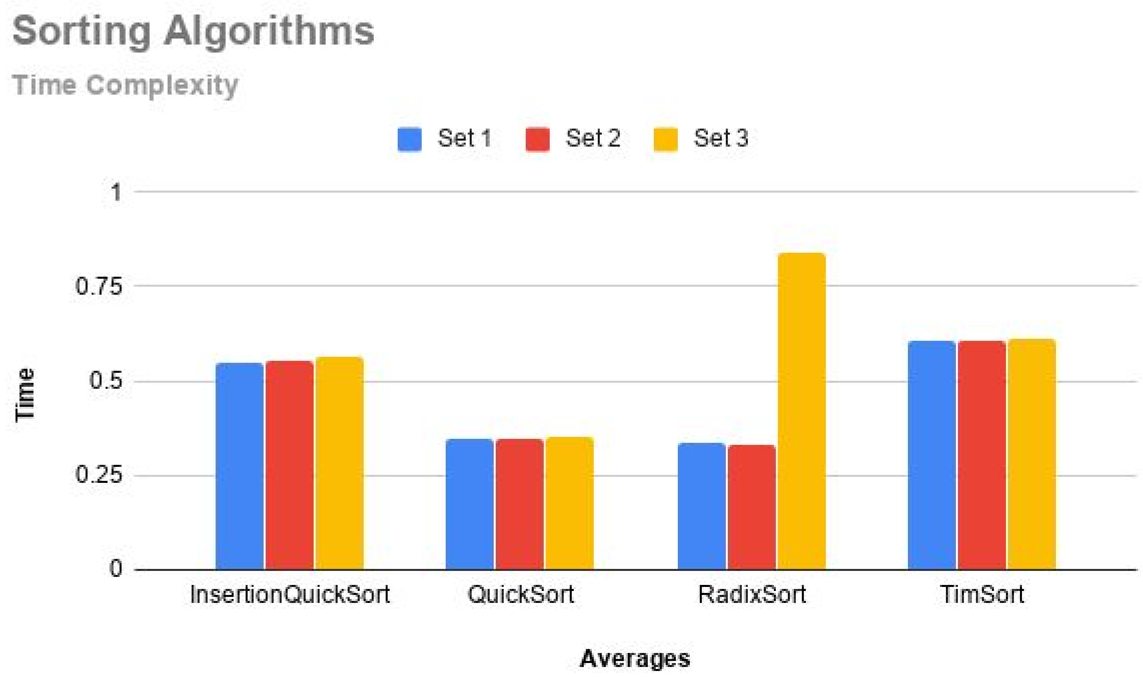
This report reports the result of an experiment that empirically tests the time complexity of sorting algorithms: InsertionQuickSort, QuickSort, RadixSort, TimSort. The experiment generates three varying types of sets each with 1 million elements; Set1: Sets where no numbers repeat, Set2: Sets where the range of numbers is 1% of the array size, Set3: Sets where no numbers repeat and each integer is of size 20. The experiment performs 10 experimental runs with each type of set on all listed sorting algorithms.

Overall TimSort had the highest time complexity, it is unexpected given that timsort is an industry standard algorithm that is adaptive to the data. The average time on each set is approximately the same for each set.

RadixSort had the highest time in set 3, which is expected given the time complexity of the algorithm $O((n + b)\ \log_b k)$; where n = number of elements, b = base, k = the number of possible values. When we have large input we have larger possible values for each element is larger, in set 3 the set has larger elements of size 20, the possible values for radix sort are between 00,000,000,000,000,000 - 99,999,999,999,999 or 100,000,000,000,000,000 possible values. Radix sort has the lowest average time in the other two sets.

Conventionally InsertionQuickSort is understood to be faster but test results show QuickSort to be faster. InsertionQuickSort is a hybrid sorting algorithm wand has varying complexities, if the data is relatively sorted in the partitions then the algorithm can take advantage of best case complexity of InsertionSort: $O(n)$, but the worst case for unsorted or in descending order data in Insertion sort is: $O(n^2)$. QuickSort has the general time complexity $\theta(nlogn)$ for sorting.

Uzair Akram



The Bar Graph Above shows the average time in seconds for each algorithm on each set.

Averages	**Set 1**	**Set 2**	**Set 3**
QuickSort	0.34628	0.3460703	0.3523064
InsertionQuickSort	0.5496419	0.5522296	0.5621179
RadixSort	0.3340907	0.3329043	0.8398559
TimSort	0.6069539	0.6054064	0.6138031

The table above shows the average time in seconds for each algorithm on each set.