CMPT 435 - Fall 2021 - Dr. Labouseur

Assignment Two

Jake Vissicchio
Jake.Vissicchio1@Marist.edu

October 18, 2021

1 Results

Sorting Magic Items		
Sort Type	Number of Comparisons Recorded	Asymtotic Running Time
Selection Sort	225415	$O(n^2)$
Insertion Sort	114927	$O(n^2)$
Merge Sort	5443	$O(n * log(n))$
Quick Sort	2975	$O(n * log(n))$

2 Explanations

Selection Sort is $O(n^2)$ because in order to sort it relies on the use of a nested for loop that we will have to go through in order to find position of the smallest value (or in this case coming first in the alphabet.

Insertion Sort is $O(n^2)$ because in order to sort it relies on the use of a nested loop. It has the potential to be faster than Selection Sort if the array is mostly sorted since it can skip more swaps and comparisons

Merge Sort is $O(n * log(n)$ because it makes use of divide and conquer. The array of n elements is repeatedly halved while sorting which results in $O(log(n))$ and then the array of n elements is merged together resulting in $O(n)$. Putting these two together we get $O(n * log(n))$.

Quick Sort is $O(n * log(n)$ because it also makes use of divide and conquer but differently from Merge Sort. While Merge Sort recursively divides until array reaches a length of one then conquers. Quick Sort recursively divides and conquers by partitioning the halves around a pivot value. By making use of a good partition value the partitioning will have a complexity of $O(n)$. With Sith Lord Partitioning such as picking the first or last value as the pivot value every time it will increase the complexity even further.
As before, the array of n elements being halved will result in $O(log(n))$. Putting these two together we get $O(n * log(n))$.