Yousef Ali

30 October 2019

CSS 342

Selection Sort vs. Bubble Sort vs. Merge Sort

**Bubble Sort:**

The bubble sort was by far the least efficient sort. This is because it goes through each number individually and switches the number of lesser value to the left and the larger number to the right. This causes a “bubble” effect as the largest number ends up on the right. The big O complexity is O(n^2).

**Selection Sort:**

The selection sort works a lot more efficiently than the bubble sort. Selection sort functions in an efficient manner because it locates the smallest number then switches it with the first number. Next, it locates the next smallest number after that and switches it into the second index and continues to do this over and over again until it reaches the end of the array. The big O complexity of the selection sort is O(n^2).

**Merge Sort:**

The merge sort is the most efficient sort out of these three. It splits the array in half continuously into temporary arrays, until it reaches a single number. It then goes back and organizes the numbers by comparing to one another which one is the least O(nlogn).

**Analysis:**

The bubble sort took around 13.3 seconds to sort 10,000 numbers and as I increased the numbers, I could see the time was increasing exponentially faster the more numbers I put into the bubble sort. This is because as there are more numbers the bubble sort needs to go through a much larger set of numbers. This is why the big O complexity is O(n^2). Due to it iterating through each number individually and repeating it over and over again the time increases when there are more numbers.

In comparison, the selection sort took about 9.3 seconds to sort 10,000 numbers. The selection sort has a similar effect as the bubble sort. The more numbers there are the more times the code needs to go back and forth checking which creates a similar bubbling type effect. Although only slightly quicker than the bubble sort, it is still nowhere near as fast as the merge sort.

As stated above the merge sort proved to be the fastest of the three. It was so quick and beating both by a massive margin. The merge sort took under 1 second to sort 10,000 numbers. According to my time code it took .005 seconds to sort all the numbers. This is a ridiculous jump from the selection sort and bubble sort. It makes a lot more sense to use the merge sort for larger data and the bubble and selection sort for smaller data because those will work completely fine and fast for small data. The merge sort is a lot quicker since it takes a lot less steps to sort the numbers and therefore a lot less work for the machine.