John Scales

Class Number 335-04

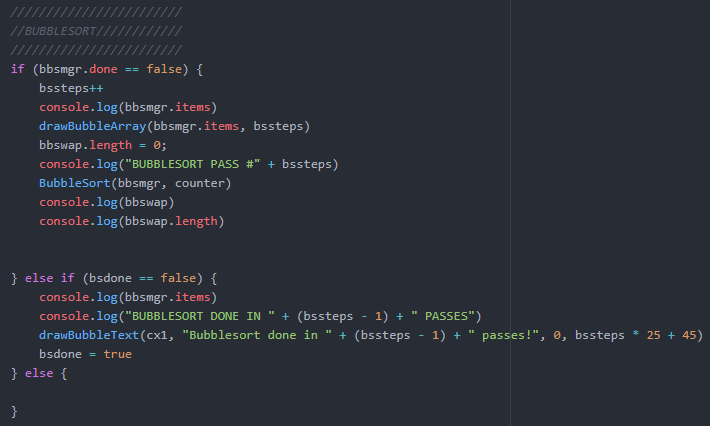
CWID: 888865730

Sort Race - Big-O Complexity

This is an interesting one to write a big O analysis on since I’m not using my algorithms, I’m just using algorithms that already exist. I did not modify them or add complexity to them. Because I didn’t add complexity, we only have to consider the worst case of the set of the worst case of the algorithms we are racing. The worst worst-case algorithms we have are Quick Sort and Bubble Sort which have a Big O complexity of O(N^2).



Here is a picture of my bubble sort algorithm which is the worst algorithm of the bunch based on Big O and Big Theta Complexities. As you can see for each pass it goes through every item in the array and swaps if necessary. In the main loop of the program we have:



This part of the bubble sort just checked if its done sorting and if it’s not it will pass to the algorithm from before and start bubbling up the next character in the list. In the worst possible case, the items will be in reverse order which will mean there will need to be N^2 swaps.

The other two algorithms don’t have as bad complexity so I will not consider them for the overall complexity of my program and are reduced to O(1).