HW 1

Christa Caggiano

26 Jan 2018

Number of Conditionals and Loops

Bubble sort

- Number of assignments: 4
- Number of conditionals 1

Quick sort

- Number of assignments: 4
- Number of conditionals: 4

![QuickSort](/Users/Christa.Caggiano/Documents/UCSF_year1/algorithms/example/Screen Shot 2018-01-26 at 11.27.25 PM.png) ![BubbleSort](/Users/Christa.Caggiano/Documents/UCSF_year1/algorithms/example/Screen Shot 2018-01-26 at 11.28.18 PM.png)

Complexity

My **Bubble sort** algorithm is $O(n^2)$ because each item in the list is accessed once in the for loop = *N* times. If the list was completely sorted, the best case, the loop will finish and the run time would be O(n). In the worst case, a completely unsorted list (ex l = [5, 4, 3, 2, 1]) the algorithm will need to go through the list *N* times in the original loop, where the 5 will bubble up. Next, bubble sort will go through the loop again until 4 bubbles up. This will continue for each number in the list, a total of *N* times. This leads to a total runtime of $O(n^2)$

Quick sort is O(n log(n)) because the original transversal of the list is *N*, needed to partition the list into two lists. With each recursive step my algorithm takes, however, the size of the list the algorithm traverses is halved, meaning each subsequent step in log(n). Thus, the overall average complexity is O(n log(n))

Github Repo

https://github.com/christacaggiano/example

Travis Build

https://travis-ci.org/christacaggiano/example