

Atiyah 'AJ' Ellerbee  
CSCI 36200  
Assignment 2

First you must understand what is going on with the selection sort and understand where misconceptions might be. I assumed initially that the worst case would be an array that was random but began with a 1, but honestly the real worst case is with an array that would be already sorted in reverse (E.G. **[5,4,3,2,1]**). The best-case scenario for the algo is an array that has already been sorted (from least to greatest ideally, E.G. **[1,2,3,4,5]**).

Both Worst-case and Best-case scenarios are quadratic. Why? Because they are both represented by the  $O(n^2)$  notation. Selection Sort will always perform the same number of comparisons regardless of whatever input order exists, and there is no early termination condition for Selection Sort. See the pseudo code below.

*For n = 5:*

*Pass 1: 4 comparisons (find min in positions 0-4)  
Pass 2: 3 comparisons (find min in positions 1-4)  
Pass 3: 2 comparisons (find min in positions 2-4)  
Pass 4: 1 comparison (find min in positions 3-4)*

The total comparisons are still the same. Overall, it would still be  $4+3+2+1$  equaling to 10 which overall is the same thing as  $n(n-1)/2$ . Let's look at the mathematical derivation for Selection Sort,  $T(n)$ , like what we went over in class.

$$\begin{aligned} T(n) &= (n-1) + (n-2) + (n-3) + \dots + 2 + 1 \\ &= \sum_{i=1}^{n-1} i \\ &= n(n-1)/2 \\ &= (n^2 - n)/2 \\ &= O(n^2) \end{aligned}$$

So going back to prove why my original thought wasn't special or wrong, if I didn't have anything besides 1 input in the array, there are no comparisons there are no swaps. This makes it  $O(1)$  notation, indicating constant time. We then assume that asymptotic behavior can be looked at  $n \rightarrow \infty$  and the quadratic growth dominates for big N. This means that most initial cases aren't important, they are trivial.

To summarize, the best-case scenario is an input with an already sorted array (E.G. **[1,2,3,4,5]**). The best-case scenario time complexity is  $O(n^2)$  and is quadratic. Sort always performs the same number of comparisons regardless of input order. It doesn't have any early termination condition. The worst-case scenario is a reverse array input (E.G.

**[5,4,3,2,1]**), but it's still quadratic because the time complexity is still  $O(n^2)$ . It's the same number of comparisons as the best case, but overall, the maximum number of swaps is used ( $n-1$  swaps). According to documentation, the structure of Selection Sort forces all non-trivial cases to be quadratic.