Programming 3 - WVA 2023-24

4/2 Homework - Question 2

Garud Shah

April 9th, 2024

1 Time Complexity

This has a time complexity of $O(N^2)$: the worst case scenario - which is when the list is in decending order, since the kth largest element will only "bubble" up and become sorted on the kth cycle. This is, since the number of comparisons on the ith cycle is $N - 1 - i$:

$$\begin{aligned} &\sum_{i=1}^{N-1} N - 1 - i && (1) \\ &= \sum_{i=1}^{N-1} i && (2) \\ &= \frac{N \cdot (N-1)}{2} && (3) \\ &= \frac{N^2}{2} - \frac{N}{2} && (4) \\ &= \boxed{O(n^2)}, && (5) \end{aligned}$$

completing the proof this algorithim is at least $O(n^2)$.
For the proof it works, which is needed to prove that our sum is valid, after $N - k - 1$ cycles it will have sorted everything as the kth largest element, which will have position $N-1-k$ in a 0-indexed list. Then forces the smallest element to the start or index 0 as all other indexes will have been taken by larger elements in sorted order, so we can safely end here. This completes the proof that the algorithim works and that our sum is valid.

2 Space Complexity

As only single comparisons and swaps are done at a time, and since the algorithim works without modification, it has $O(1)$ space complexity.

3 Algorithim Grades

Note: this is question 3.