1. The worst case occurs when the largest or smallest element is always chosen for the pivot
    a. Results in a sub-array of size 0, and another of size $n - 1$
    b. Is $O(n^2)$
2. Let's consider a scenario where we always pick the last element as the pivot. A vector that incurs the worst-case complexity in this situation would be one that is already sorted in ascending order, as the pivot (being the last element) will always be the largest in the sub-array.

    Vector = **[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]**

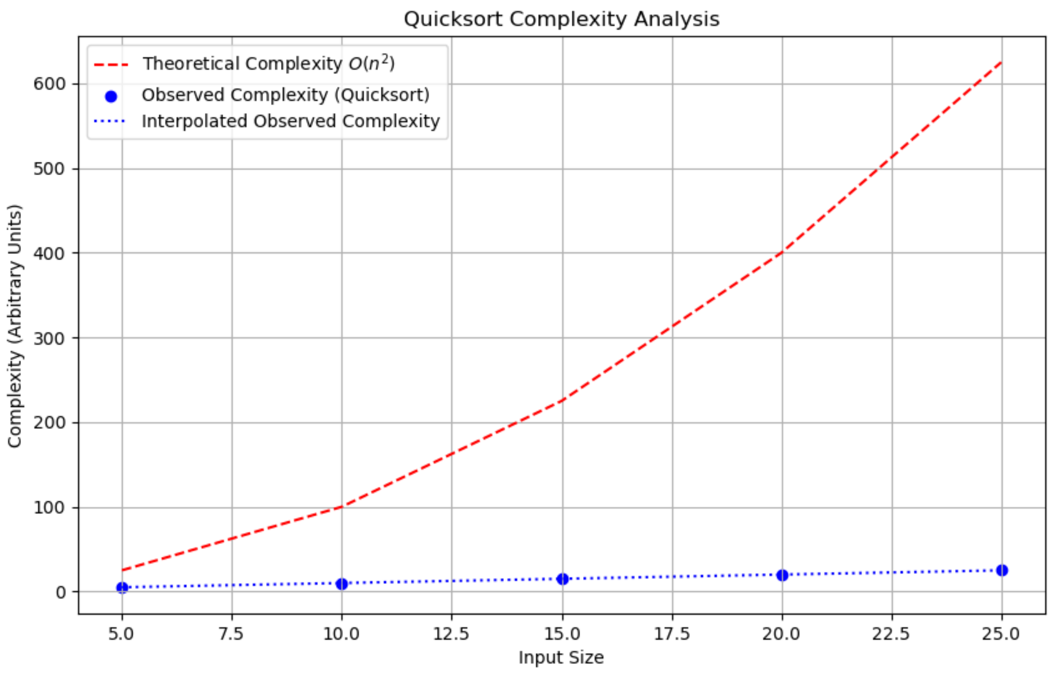
    Now, let's manually show the working of quicksort on this vector:

    1. **First Pass (Pivot = 16):**
        i. Elements less than 16: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
        ii. Elements greater than 16: []
        iii. New Vector: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, |16|]
    2. **Second Pass (Pivot = 15):**
        i. Elements less than 15: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
        ii. Elements greater than 15: []
        iii. New Vector: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, |15|, 16]
    3. **Third Pass (Pivot = 14):**
        i. And so on, until each element has been "sorted".

    In each pass, the pivot is placed in its correct position (indicated by the vertical bar | |), but since it's always the largest element in the sub-array, we end up having to sort the rest of the elements in a very inefficient manner. Each step reduces the problem size by only 1, leading to n-1 recursive calls, which is the worst-case scenario.

    This process continues, with each step sorting a sub-array that is one element smaller than the previous sub-array, until each element is in its correct place. This results in $O(n^2)$ time complexity due to the $n+(n-1)+(n-2)+\ldots+1$ sequence of operations, where n is the number of elements in the vector.

4.


Quicksort Complexity Analysis
Theoretical Complexity $O(n^2)$
Observed Complexity (Quicksort)
Interpolated Observed Complexity
Complexity (Arbitrary Units)
Input Size
0
100
200
300
400
500
600
5.0
7.5
10.0
12.5
15.0
17.5
20.0
22.5
25.0