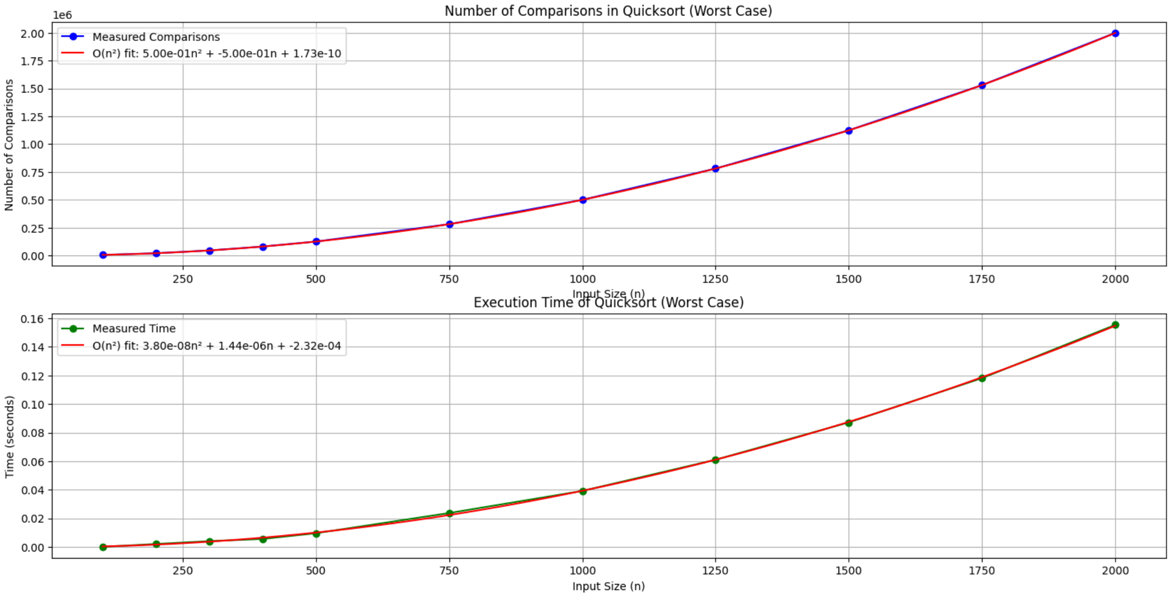
EX4

4.



1.- Let T(n) be the time complexity for sorting an array of size n

```
    - In the worst case, each partition operation divides the array into subarrays of size 0 and n-1
    - Each partition operation itself takes O(n) time (to compare each element with the pivot)

    This gives us the recurrence relation:
    T(n) = T(n-1) + T(0) + O(n)

    Since T(0) = 0, this simplifies to:
    T(n) = T(n-1) + O(n)

    Expanding this recursion:
    T(n) = T(n-1) + O(n)
          = T(n-2) + O(n-1) + O(n)
          = T(n-3) + O(n-2) + O(n-1) + O(n)
          ...
          = T(1) + O(2) + O(3) + ... + O(n-1) + O(n)
          = O(1) + O(2) + O(3) + ... + O(n-1) + O(n)
          = O(1 + 2 + 3 + ... + n-1 + n)
          = O(n(n+1)/2)
          = O(n²)

    Therefore, the worst-case time complexity of quicksort is O(n²).
```

3.Derivation of Worst-Case Complexity for Quicksort:

```
    In the worst case scenario for quicksort:
    1. The pivot chosen at each step results in the most unbalanced
partition possible
    2. With last-element pivot selection, this occurs when the array is
already sorted
```