1) deriving worst case complexity.

Letting $C(n)$ to denote the number of comparisons, required to sort an array of size n. (in worst case).

Then:

$$C(n) = C(n-1) + (n-1) + C(0)$$

- $C(n-1)$: comparisons in partioning step.
- $(n-1) + C(0)$: worst case scenario (assuming pivot is either smallest / largest element).

So

$$C(n) = C(n) + (n-1)$$

$$= [C(n-2) + (n-2)] + (n-1)$$

$$= C(n-3) + (n-2) + (n-1)$$

$$\vdots$$

$$= C(1) + \underbrace{1 + 2 + 3 + \cdots (n-1)}$$

$$= C(1) + \frac{n(n-1)}{2}$$

$C(1)$: Base case.

Big O notation

Hence $\longrightarrow$ $C(n) = \frac{n(n-1)}{2} = O(n^2)$

2) consider vector array:
in descending order. Consider vector alpha as a reverse sorted array:

Alpha: [33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18]

Step 1: we choose extremely unoptimized pivot seletion. considering our first pivot element is right most.

First Partitioning (Pivot at [18])

old state:

[33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19] | [18]

new state:

[18] | [33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19]

second Partition (Pivot at = [19]

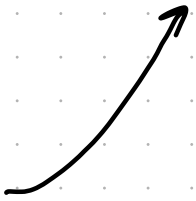
[18, 19] [33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20,]

we will use succcuive steps for 20, 21, 22, ...

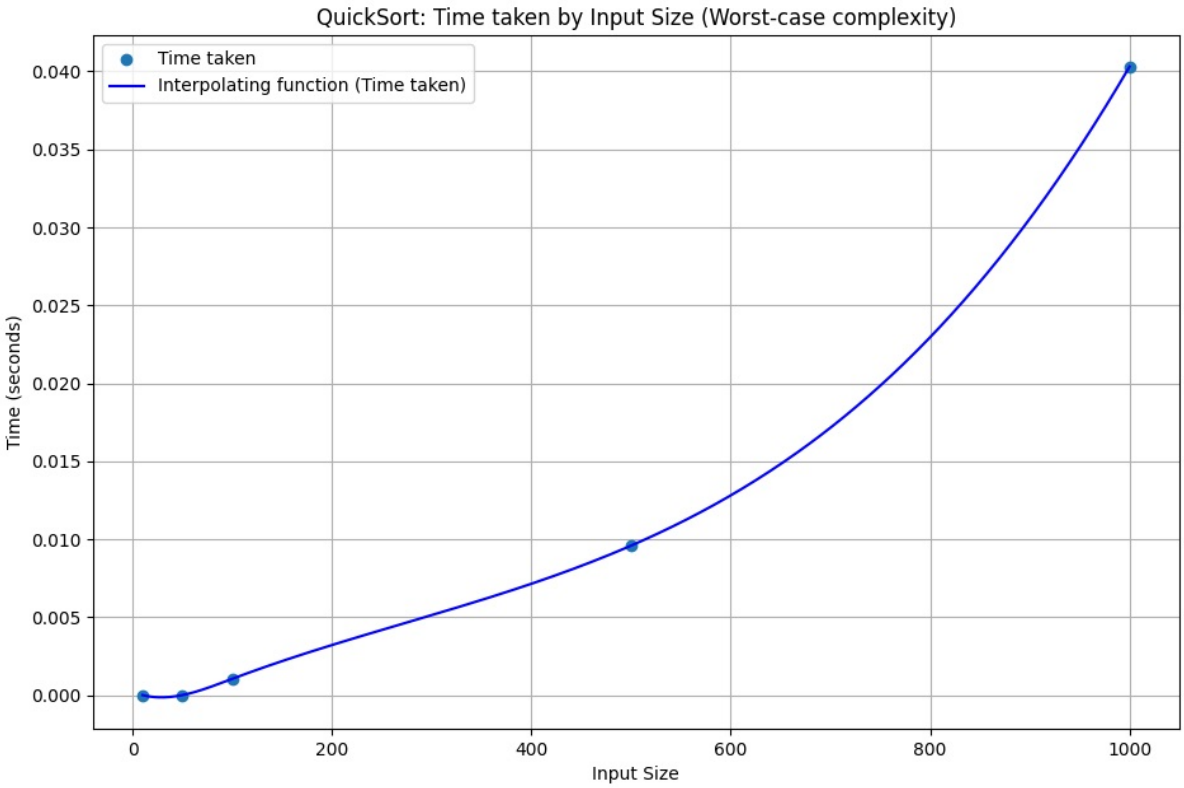
we will apply till Final step (pivot = 33)

Finally:

[18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33]



Big O $\longrightarrow$ $O(n^2)$. (had to entire array during the traversal). giving worst case.



4) yes the plot precisely follows the discussed $O(n^2)$ complexity. If we plot n^2 in the desmos, we will have similar quadratic curve that we see in the implementation above.