1) derriving worst case complexity.

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

Then:

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

comparisons in partionting step.

worst case scenario (assuming pivot is either smallest / largest element).

So

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

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

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

$$\vdots$$

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

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

Base case.

Big O notation

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

2) consider vector array:
in ascending order.

$V = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]$

1) selecting pivot 16.

As its rightmost, and all other ellements are less than pivot. The pivot is sorted right most.

Similarly we have partitioned it in left 15 elements and right 16^{th} element.

Left partitio   Pivot.

so

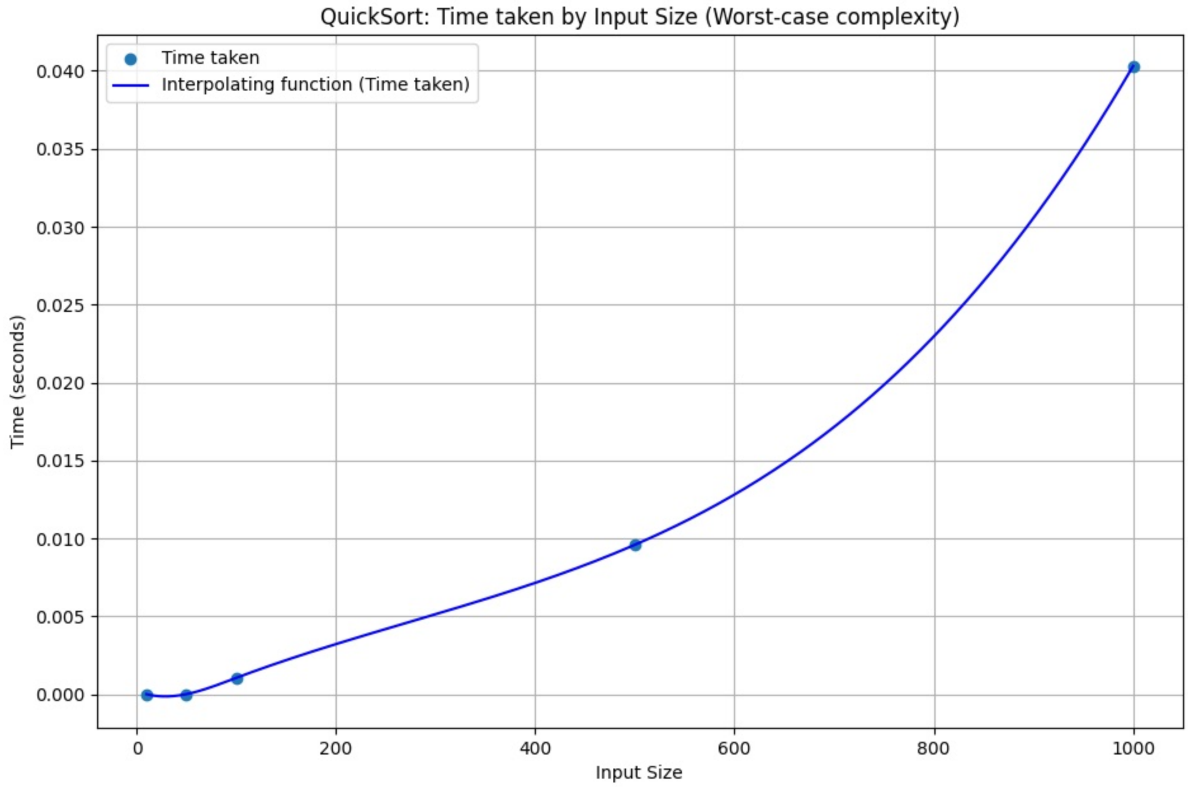
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15] [16]

Following this.

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

This alogorithm will be applied successively on the left mot partition giving us.

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15]



4) yes the plot precisely follows the discussed $O(n^2)$ complexity.