Problems on QuickSort

1) Sorting of n numbers → $n/7$ th smallest element is selected as pivot using $O(n^2)$ time complexity.

⇓

worst case time complexity of quicksort

0	1	2	3	4	5	6
70,	50,	45,	79,	82,	65,	95
1	2	3	4	5	6	7

→ 3rd smallest element as a pivot

⇓

3rd Position after applying Partition algo

$n/7$ th smallest element

⇓

$n/7$ Position

$$\underbrace{50, 65}\,,\ \overset{3}{\underline{70}}\,,\ \underbrace{\quad}$$

Pivot Element

$$\underbrace{\qquad\qquad\qquad}_{n/7 - 1}\ \ \underline{\frac{n}{7}}\ \ \underbrace{\qquad\qquad\qquad\qquad}_{n - n/7}$$

chosen pivot ⇑

Recurrence Relation

$$T(n) = \underbrace{T\left(\frac{n}{7} - 1\right)}_{\text{Left subarray}} + \underbrace{T\left(n - \frac{n}{7}\right)}_{\text{Right subarray}} + \underline{n^2} +$$

n ⇓ Partition

$$T(n) \Rightarrow T\left(\frac{n}{7}\right) + T\left(\frac{6n}{7}\right) + n^2$$ ⇒ Recursive Tree Approach

$$\Rightarrow \underline{O(n^2)}$$

2) Sorting of n numbers → $n/10$ th element selected as pivot using $O(\log n)$. (Not mentioning about smallest or largest)

⇓

worst case time complexity of quicksort

$$T(n) = T(n-1) + \underline{\log n} + \underline{n}$$

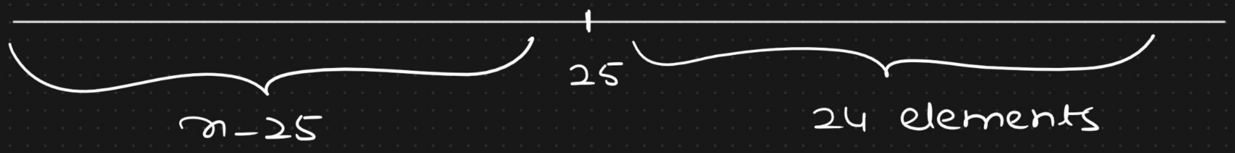
$$T(n) = T(n-1) + n$$

$$\boxed{T(n) = O(n^2)}$$

3) Sorting of n numbers ⇒ 25th Largest element is selected as pivot using $O(n^2)$

⇓

worst case time complexity of QuickSort



$$T(n) \Rightarrow T(n-25) + T(24) + \underline{n^2 + n}$$

$$T(n) \Rightarrow T(n-25) + n^2$$

$$\boxed{T(n) \Rightarrow O(n^3)}$$