Best Case

As every recursion will make 3 recursive calls with $n/3$ data set, the time complexity :

$T(n) = 3T(n/3) + O(n)$

Using Master Method:

$a = 3$

$b = 3$ $\quad\quad f(n) = \Theta(n^{\log_b a})$

$f(n) = O(n)$ $\quad\quad n = n$ (as $\log_3 3 = 1$)

Therefore $T(n) = \Theta(n \log n)$

Worst Case

As the first and third partition contains 0 elements, & partition two contains n-2 elements and it iterates once (through the array) we can write :

$T(n) = T(n-2) + O(n)$

$= = T(n-2) + n$

It will be $O(n^2)$

⟸

Using Substitution:

$T(n) = T(n-4) + O(n-2) + O(n)$

$= T(n-6) + O(n-4) + O(n-2) + O(n)$

$= T(n-8) + O(n-6) + O(n-4) + O(n-2) + O(n)$

k times