Insertion Sort

worst case complexity: O (n square)
Suppose, an array is in ascending order, and you want to sort it in descending order, In the case, worst case complexity occurs.
Each element has to be compared with each of the other element. So for every n th element (n-1) number of comparisons are made.
The total number of comparison = $n(n-1) \sim n(2)$

Best case complexity: O(n) when the array is already sorted the outer loop runs for n number of times whereas the inner loop does not run at all. So there are only n number of comparisons. Thus, complexity is linear.

Average case complexity: $O(n^2)$
It occurs when the elements of an array are in jumbled order (neither ascending nor descending)

Selection Sort

Selecting the lowest element requires scanning all n elements ($n-1$) comparisons and then swapping it into the first position.

Finding the next lowest element requires scanning the remaining ($n-1$) elements and so on.

$= (n-1) + (n-2) + \cdots + 2 + 1 = n(n-1)/2$

$= O(n^2)$ comparisons

Best case: $O(n)^2$

Worst case: $O(n)^2$

Average case: $O(n)^2$