

Average case:

In Bubble Sort, $n-1$ comparisons are done in the 1st pass, $n-2$ in the 2nd pass, $n-3$ in the 3rd pass until the array is sorted.

The total number of comparisons is:

$$(n-1) + (n-2) + (n-3) + \dots + 3 + 2 + 1 \\ = \sum_{i=0}^{n-1} n(n-1)/2 = O(n^2)$$

This results in the average case time complexity of $O(n^2)$.

Best case:

The best case for Bubble Sort is when the array is already sorted.

In this case, the algorithm checks N number of elements to see if swapping adjacent elements is needed.

Therefore, the best case time complexity of Bubble Sort is $O(n)$.

Worst case:

The worst case for Bubble Sort is when the array is sorted in reverse order.

The same number of comparisons is done as the Average case, where $n-1$ comparisons are done in the 1st iteration, $n-2$ in the 2nd iteration, and so on.

The total number of comparisons is:

$$(n-1) + (n-2) + (n-3) + \dots + 3 + 2 + 1 \\ = \sum_{i=0}^{n-1} n(n-1)/2 = O(n^2)$$

This results in the worst case time complexity of $O(n^2)$.