The time complexity of the Insertion Sort algorithm in both the worst case and the best case scenarios is as follows:

* Worst case time complexity: O(n^2) In the worst case scenario, when the input array is sorted in descending order, each element needs to be compared to all the previous elements, resulting in a total of n\*(n-1)/2 comparisons, where n is the size of the input array. This makes the worst case time complexity of the Insertion Sort algorithm O(n^2).
* Best case time complexity: O(n) In the best case scenario, when the input array is already sorted in ascending order, each element only needs to be compared to its predecessor and no swapping is required. This results in a total of n-1 comparisons, making the best case time complexity of the Insertion Sort algorithm O(n).

Overall, the Insertion Sort algorithm has a quadratic time complexity in the worst case scenario and a linear time complexity in the best case scenario. However, in practice, the worst case scenario is rare and the Insertion Sort algorithm is often used for small to medium sized arrays, where its simplicity and stability make it an efficient sorting algorithm.