**10675853**

**Insertion sort Algorithm**

The insertion sort is used when:

* the array is has a small number of elements
* there are only a few elements left to be sorted

**Time Complexities**

* **Worst Case Complexity:**O(n2)  
  A worse case complexity occurs when an array is in ascending order, and you want to sort it in descending order. Each element has to be compared with each of the other elements so, for every nth element, (n-1) number of comparisons are made.  
    
  Thus, the total number of comparisons = n\*(n-1) ~ n2
* **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 is only n number of comparison. Thus, complexity is linear.
* **Average Case Complexity:**O(n2)  
  It occurs when the elements of a array are in jumbled order (neither ascending nor descending).

**Space Complexity**

**s**Space complexity is O(1) because an extra variable key is used.

**How does insertion sort work?**

The first element in the array is assumed to be sorted. Take the second element and store it separately in key.  
  
Compare key with the first element. If the first element is greater than key, then key is placed in front of the first elemet.

The first element in the array is assumed to be sorted. Take the second element and store it separately in key.  
  
Compare key with the first element. If the first element is greater than key, then key is placed in front of the first elemet.

In a similar way, place every unsorted element at its correct position.