**Insertion Sort**

This is a in-place comparison based sorting algorithm. Here, a sub-list is maintained which is always sorted. For example, the lower part of an array is maintained to be sorted. A element which is to be 'insert'ed in this sorted sub-list, has to find its appropriate place and insert it there. Hence the name **insertion sort**.

The array is searched sequentially and unsorted items are moved and inserted into sorted sub-list (in the same array). This algorithm is not suitable for large data sets as its average and worst case complexity are of Ο(n2) where n are no. of items.

**Algorithm:**

Now we have a bigger picture of how this sorting technique works, so we can derive simple steps by which we can achieve insertion sort.

Step 1- If it is the first element, it is already sorted. Return 1;

Step 2- Pick next element

Step 3 - Compare with all elements in the sorted sub-list

Step 4 - Shift all the elements in the sorted sub-;ist that is greater than the value to be sorted

Step 5 - Insert the value

Step 6 - Repeat until the list is sorted