INSERTION SORT ALGORITHM

I/ DEFINITION:

- This is an 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. An element which is to be 'insert'ed in this sorted sub-list, has to find its appropriate place and then it has to be inserted there. Hence the name, **insertion sort**.

- The array is searched sequentially and unsorted items are moved and inserted into the 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 $O(n^2)$, where **n** is the number of items.

==

II/ PSEUDOCODE IMPLEMENTATION:

1. If the first element already sorted. Return 1.

2. Pick the next element.

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

4. Shift all the elements in the sorted sub-list that is greater than the value to be sorted.

5. Insert the value.

6.Repeat until list is sorted.

```
procedure insertionSort( A : array of items )
   int holePosition
   int valueToInsert
   for i = 1 to length(A) inclusive do:
   // select value to be inserted
      valueToInsert = A[i]
      holePosition = i
  //locate hole position for the element to be inserted
      while holePosition > 0 and A[holePosition-1] >
valueToInsert do:
         A[holePosition] = A[holePosition-1]
         holePosition = holePosition -1
      end while
   // insert the number at hole position
      A[holePosition] = valueToInsert
   end for
end procedure
```