

## Problem 5 – Insertion Sort

- **Pseudocode**

INSERTION-SORT (*A*)

```
1 for j <- 2 to length[A]  
2   do key <- A[j]  
3     Insert A[j] into the sorted sequence A[1 .. j - 1].  
4     i <- j - 1  
5     while i > 0 and A[i] > key  
6       do A[i + 1] <- A[i]  
7       i <- i - 1  
8     A[i + 1] <- key
```

- **The mechanism of insertion sort**

with the result as the first element of the input.

Loop over the input until it is empty, "removing" the first remaining (leftmost) element.

Compare the removed element against the current result, starting from the highest (rightmost) element, and working left towards the lowest element.

If the removed input element is lower than the current result element, copy that value into the following element to make room for the new element below, and repeat with the next lowest result element.

Otherwise, the new element is in the correct location; save it in the cell left by copying the last examined result up, and start again from (2) with the next input element.

- **Stable sorting algorithms :-**

Is an algorithm that maintains the relative order of items with equal values .

- Insertion sort is a stable algorithm
- Time complexity
  - Worst case  $\rightarrow O(n^2)$
  - Best case  $\rightarrow O(n)$