

Sorting Algorithm – Insertion Sort

Insertion sort builds the sorted list by examining one number at a time from beginning to end, then rearranging it if necessary by placing it in the correct position in the sequence of numbers before it. After each number is examined and rearranged, the size of the sorted list grows by one.

Suppose the following list is to be sorted in ascending order. Numbers will be examined from left to right. First, 9 is examined. Note that 25 does not need to be checked since it is already in the correct position in the list of size one.

25	9	11	31	23	2	42	15	26
----	---	----	----	----	---	----	----	----

Since $9 < 25$, the 9 should be inserted to the left of 25, which produces the list in which the first two values on the left are in order (a sorted list of size 2).

9	25	11	31	23	2	42	15	26
---	----	----	----	----	---	----	----	----

The next number 11 should be inserted between 9 and 25, giving the next arrangement in which the first three numbers on the left are in order (a sorted list of size 3).

9	11	25	31	23	2	42	15	26
---	----	----	----	----	---	----	----	----

Next, 31 is examined. Since it is greater 25 (which means it is greater than all values to its left), it should remain where it is. Now, the first four values on the left are in order (a sorted list of size 4).

9	11	25	31	23	2	42	15	26
---	----	----	----	----	---	----	----	----

All subsequent numbers are examined one by one and placed in the correct position until the entire list is sorted.

9	11	23	25	31	2	42	15	26
---	----	----	----	----	---	----	----	----

2	9	11	23	25	31	42	15	26
---	---	----	----	----	----	----	----	----

2	9	11	23	25	31	42	15	26
---	---	----	----	----	----	----	----	----

2	9	11	15	23	25	31	42	26
2	9	11	15	23	25	26	31	42

Implementation

To implement the insertion sort that sorts a list in ascending order, each item in the list will be compared to all the items before it. If the item is smaller than the items before, those items will be shifted until the item can be inserted to create a list in order.

```

insertionSort(List L)
  for each item x in the list
    set j to the index of x
    while j is greater than 0 and x is less than the item at index (j-1)
      set value at index j to item at index (j-1)
      set j to j - 1
    set value at index j to item x

```