

## The Operation Of Insertion Sort

Insertion sort is an interesting sort that is slightly non-intuitive. Most people who try to come up with an algorithm for sorting usually come up with *selection sort* on their first attempt. Insertion sort is slightly more efficient, in general.

Insertion sort, as defined in the text, starts with the second element in the sequence, using the fact that a sequence of length 1 is always sorted. Insertion sort then checks the second element and places it in the correct order in the first 2 elements. It then moves on to the third element and inserts that into the correct place in the first three elements. Note that the sorted part of the sequence always starts at the head of the sequence and grows towards the tail, whereas selection sort has the sorted portion at the tail and it grows towards the head. Also note that in insertion sort the sorted portion of the sequence is not necessarily in final order, whereas in selection sort it is.

### Example 1

Suppose we wish to sort the sequence below, using insertion sort.

P	M	S	C	F	I
---	---	---	---	---	---

This is the sequence before sorting starts, remembering that a sequence of length 1 is always sorted.

<b><i>P</i></b>	M	S	C	F	I
-----------------	---	---	---	---	---

← The bold italic portion is sorted but may not be in final position yet.

After sorting the second element:

<b><i>M</i></b>	<b><i>P</i></b>	S	C	F	I
-----------------	-----------------	---	---	---	---

After sorting the third element:

<b><i>M</i></b>	<b><i>P</i></b>	<b><i>S</i></b>	C	F	I
-----------------	-----------------	-----------------	---	---	---

The rest of the sort proceeds as follows.

<b><i>C</i></b>	<b><i>M</i></b>	<b><i>P</i></b>	<b><i>S</i></b>	F	I
<b><i>C</i></b>	<b><i>F</i></b>	<b><i>M</i></b>	<b><i>P</i></b>	<b><i>S</i></b>	I
<b><i>C</i></b>	<b><i>F</i></b>	<b><i>I</i></b>	<b><i>M</i></b>	<b><i>P</i></b>	<b><i>S</i></b>

In contrast selection sort finds the largest item in the whole sequence and moves it to the back. It then finds the next largest item and moves that to the second to last position, etc.

Before starting, we have:

P	M	S	C	F	I
---	---	---	---	---	---

After one item is sorted, we have:

P	M	I	C	F	<b>S</b>
---	---	---	---	---	----------

Note that the largest item is *swapped* with the item at the back. The bold italic portion is sorted *and* in final position

Next we have:

F	M	I	C	<b>P</b>	<b>S</b>
---	---	---	---	----------	----------

Continuing on in this way, we find the sort proceeds as follows.

F	C	I	<b>M</b>	<b>P</b>	<b>S</b>
F	C	<b>I</b>	<b>M</b>	<b>P</b>	<b>S</b>
C	<b>F</b>	<b>I</b>	<b>M</b>	<b>P</b>	<b>S</b>
<b>C</b>	<b>F</b>	<b>I</b>	<b>M</b>	<b>P</b>	<b>S</b>

The last element is actually sorted as soon as the second to last item is sorted, but the algorithm will not be able to tell this.

◆◆◆

Note that in insertion sort there is no part of the sequence that is guaranteed to be in its final position until after the last item is sorted. By contrast, in selection sort, there is always guaranteed to be some portion of the sequence in final position after the sort is started.

### Example 2

Use the table below to illustrate both sorts on the sequence L, E, K, Y, B, N. Remember that you will be showing the results after each recursive call to selection sort and after each return in insertion sort. The first line has been done for you.

Selection Sort						Insertion Sort					
L	E	K	Y	B	N	L	E	K	Y	B	N
L	E	K	N	B	Y	E	L	K	Y	B	N

◆◆◆