

**ALGORITHM 2.2** Insertion sort

```

public class Insertion
{ // Insertion sort.
  public static void sort(Comparable[] a)
  { // Sort a[] into increasing order.
    int N = a.length;
    for (int i = 1; i < N; i++)
    { // Insert a[i] among a[i-1], a[i-2], a[i-3]... ..
      for (int j = i; j > 0 && less(a[j], a[j-1]); j--)
        exch(a, j, j-1);
    }
  }
  // See page 151 for less(), exch(), and main().
}

```

For each  $i$  from 0 to  $N-1$ , exchange  $a[i]$  with the entries that are smaller in  $a[0]$  through  $a[i-1]$ . As the index  $i$  travels from left to right, the entries to its left are in sorted order in the array, so the array is fully sorted when  $i$  reaches the right end.

		a[]										
i	j	0	1	2	3	4	5	6	7	8	9	10
		S	O	R	T	E	X	A	M	P	L	E
1	0	<b>O</b>	S	R	T	E	X	A	M	P	L	E
2	1	0	<b>R</b>	S	T	E	X	A	M	P	L	E
3	3	0	R	S	<b>T</b>	E	X	A	M	P	L	E
4	0	<b>E</b>	O	R	S	T	X	A	M	P	L	E
5	5	E	O	R	S	T	<b>X</b>	A	M	P	L	E
6	0	<b>A</b>	E	O	R	S	T	X	M	P	L	E
7	2	A	E	<b>M</b>	O	R	S	T	X	P	L	E
8	4	A	E	M	O	<b>P</b>	R	S	T	X	L	E
9	2	A	E	<b>L</b>	M	O	P	R	S	T	X	E
10	2	A	E	<b>E</b>	L	M	O	P	R	S	T	X
		A	E	E	L	M	O	P	R	S	T	X

*entries in gray  
do not move*

*entry in red  
is a[j]*

*entries in black  
moved one position  
right for insertion*

Trace of insertion sort (array contents just after each insertion)