

ALGORITHM 2.1 Selection sort

```

public class Selection
{ // Selection sort.
  public static void sort(Comparable[] a)
  { // Sort a[] into increasing order.
    int N = a.length;           // Array length.
    for (int i = 0; i < N; i++)
    { // Exchange a[i] with smallest entry in a[i+1..N].
      int min = i;              // index of minimal entry.
      for (int j = i+1; j < N; j++)
        if (less(a[j], a[min])) min = j;
      exch(a, i, min);
    }
  }
  // See page 151 for less(), exch(), and main().
}

```

For each i , this implementation puts the i th smallest entry in $a[i]$. The entries to the left of position i are the i smallest entries in the array and are not examined again.

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

entries in black are examined to find the minimum
entries in red are a[min]
entries in gray are in final position

Trace of selection sort (array contents just after each exchange)