

Selection Sort Algorithm

This is to help you with programming exercise 3, on page 37: the pseudocode of Selection Sort

```
procedure SelectionSort( $a_1, a_2, \dots, a_n$ ):
    sorted = 0 # index of sorted elements
    while sorted < n-1: # we need to sort the first n-1 elements only
        m = FindSmallest( $a_{sorted}, \dots, a_n$ ) # find next smallest
        # note that m is a relative position, in the sequence [ $a_{sorted}, \dots, a_n$ ]
        # swapping the next min value with the element at current position
        tmp =  $a_{sorted}$ 
         $a_{sorted} = a_{m+sorted}$ 
         $a_{m+sorted} = tmp$ 

        sorted += 1 # increment current position index, the number of
sorted elements in the sequence.
```

We need to define the procedure of finding the smallest element in the list. It can be done by a built-in function call, or by defining a new function/procedure/method.

```
procedure FindSmallest( $a_1, a_2, \dots, a_n$ ):

    s = 0 # index of smallest element
    i = 0 # index/counter
    while i < n: # check all the elements in the list
        if  $a_s > a_i$ : s = i
        i +=1
    return s
```

This procedure returns the index of the smallest element in the sequence a_1, a_2, \dots, a_n .

Another option is to use the built-in **min** method. This will yield a better running time!