**function** *insertionSort*(array A)

**for** i **from** 1 **to** length[A]-1 **do**

value := A[i]

j := i-1

**while** j >= 0 **and** A[j] > value **do**

A[j+1] := A[j]

j := j-1

**done**

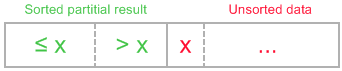
A[j+1] = value

**done**

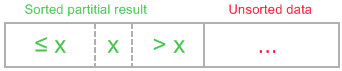
OR

**Algorithm**

Insertion sort algorithm somewhat resembles [selection sort](http://www.algolist.net/Algorithms/Sorting/Selection_sort). Array is imaginary divided into two parts - **sorted one** and**unsorted one**. At the beginning, **sorted part** contains **first element** of the array and **unsorted one** contains the rest. At every step, algorithm takes **first element** in the **unsorted part** and **inserts** it to the right place of the**sorted one.** When**unsorted part** becomes **empty**, algorithm *stops*. Sketchy, insertion sort algorithm step looks like this:



**becomes**



The idea of the sketch was originaly posted [here](http://en.wikipedia.org/wiki/Insertion_sort).

Let us see an example of insertion sort routine to make the idea of algorithm clearer.

*Example.*Sort {7, -5, 2, 16, 4} using insertion sort.

