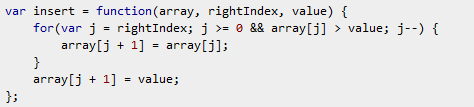
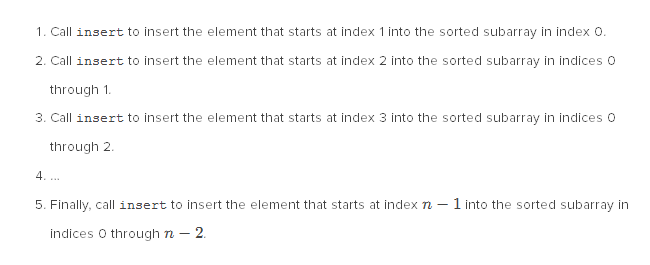
It is like if you had one card in your hand and the dealer was giving you new cards, one at a time, and you loop everytime through your hand to see where the new card has to go. So we are going the other way around. In selection sort, we were starting at index 0 and checking all the cards to the RIGHT to see whether ours was the smallest one. Now, we pick a card at index i+1 and check whether the cards on the LEFT (which are already sorted) are smaller, and we put then the new card where it belongs. We slide the card to the LEFT continuously until it finds its place.

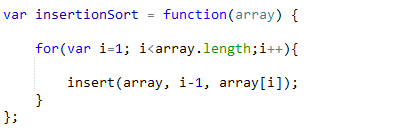
Here we implement an insert function that inserts a value to the right of a sorted subarray and then loops from left to right, and if the value of array[i] is bigger than the value, it exchange the position of value and array[i], so value *slides* to the right until the next number is smaller than value. Notice how they construct the for loop, in the second condition, array[j] >= value is implemented.



In order to sort a whole unsorted array, we simply loop that function starting at index 0 (because one value can be considered as an array of length one, and will by definition be sorted).



The whole insertion sorting algorithm looks like this:



Be sure to specify i-1 and not I, otherwise the insert function will never loop (since value = array[i] and the condition array[j] < value we just talked about has to be fulfilled for the loop to begin)

The big-theta here is variable, it can be theta(n²) if we have to sort everything in an array or theta(n) if it’s already partly sorted.