Example of sorting

Algorithm: Insertion Sort

This algorithm sorts the sequence $s_1, \ldots, s_n$ in nondecreasing order.

Input: s, n

Output: s (sorted)

```
insertion_sort(s, n) {
   for i = 2 to n {
      val = $s_i$ // save $s_i$ so it can be inserted into the correct place
      j = i - 1
      // if val < $s_j$, move $s_j$ right to make room for $s_i$
      while ($j \geq 1 \land val < s_j$) {
            $s_{j+1} = s_j$
            j = j – 1
      }
      $s_{j+1}$ = val // insert val
   }
}
```

1. val is used as a "temp" to remember the value of the number we are trying to sort.
2. j is a moving index to find the right place for Si.
3. In the "while" loop, we trying to move all the number that is bigger than Si to its right, so we can have space for the Si to put in, also we don't need to worry about Si getting replaced because we put it into the "temp".
4. We stop this when we find a Sj which is smaller than/equal to Si, then we put Si on the right of it. The worst case for each iteration is that when Si is the smallest number, so the loop stops at "j=0" then we put Si at the first spot (0+1).
5. For the for loop, we sort every Si from the second one to the end, one by one.

More Information: https://www.youtube.com/watch?v=DFG-XuyPYUQ

Exercise of Insertion sort: Trace the algorithm of the Insertion Sort (slide 18) for the input

34 20 144 55

First, 20 is inserted in

34

Since 20 < 34, 34 must move one position to the right

	34

Now 20 is inserted

20	34

Since 144 > 34, it is immediately inserted to 34’s right

20	34	144

Since 55 < 144, 144 must move one position to the right

20	34		144

Since55 > 34, 55 is now inserted

20	34	55	144

The sequence is now sorted.