INSERTION SORT

Insertion Sort

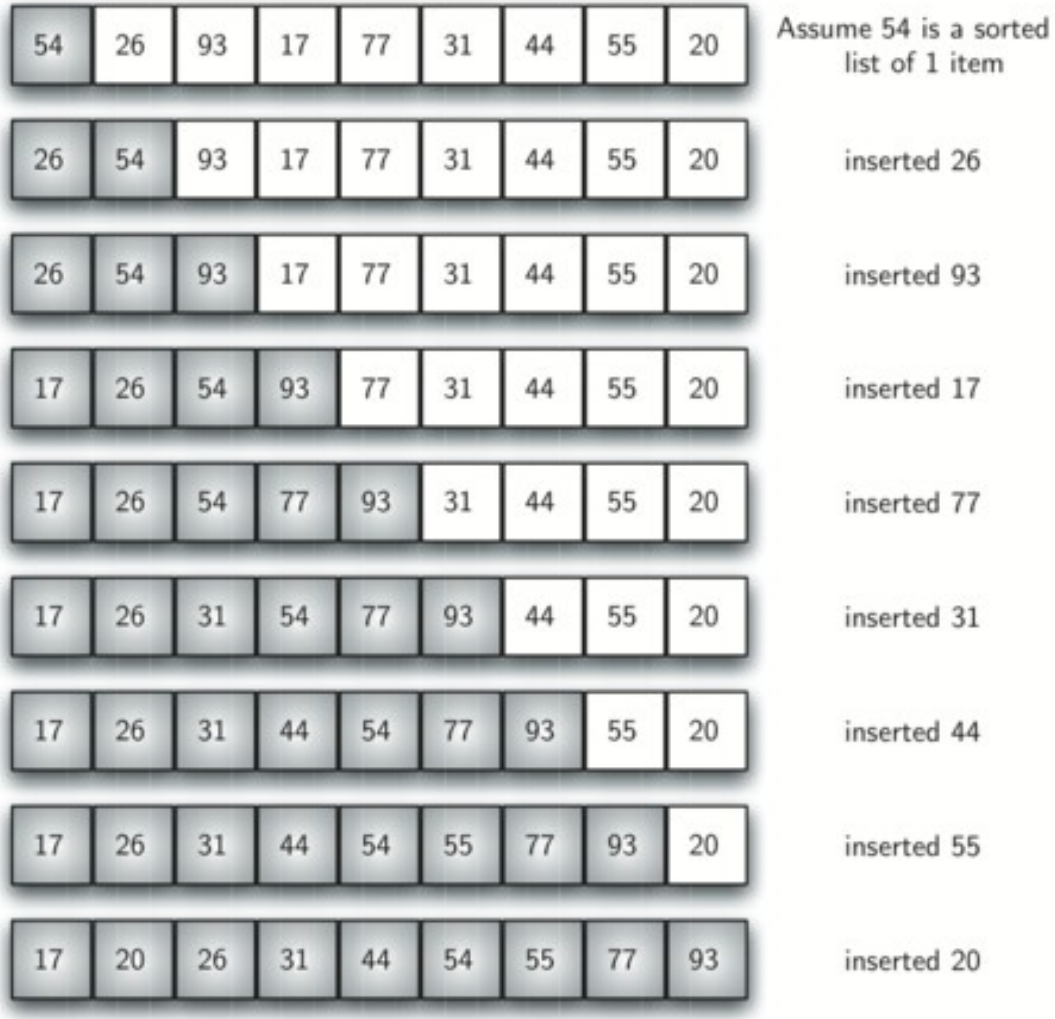
- Explanation of Insertion Sort
- Visualization of Insertion Sort
- Implementation of Insertion Sort

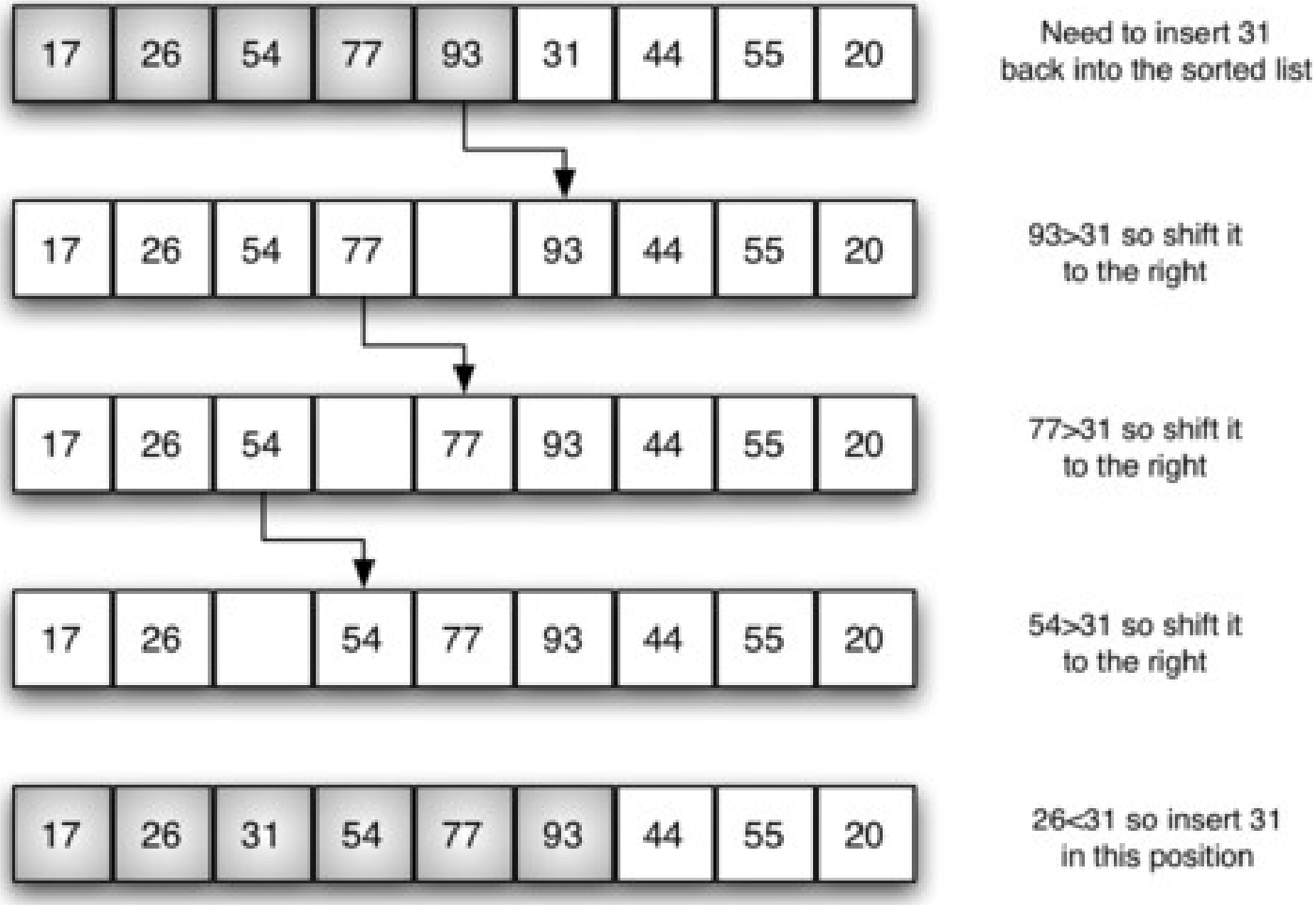
Insertion Sort

- The **insertion sort** always maintains a sorted sublist in the lower positions of the list.
- Each new item is then “inserted” back into the previous sublist such that the sorted sublist is one item larger.

Insertion Sort

- We begin by assuming that a list with one item (position 0) is already sorted.
- On each pass, one for each item 1 through n−1, the current item is checked against those in the already sorted sublist.
- As we look back into the already sorted sublist, we shift those items that are greater to the right.
- When we reach a smaller item or the end of the sublist, the current item can be inserted.


54 26 93 17 77 31 44 55 20
Assume 54 is a sorted list of 1 item
26 54 93 17 77 31 44 55 20
inserted 26
26 54 93 17 77 31 44 55 20
inserted 93
17 26 54 93 77 31 44 55 20
inserted 17
17 26 54 77 93 31 44 55 20
inserted 77
17 26 31 54 77 93 44 55 20
inserted 31
17 26 31 44 54 77 93 55 20
inserted 44
17 26 31 44 54 55 77 93 20
inserted 55
17 20 26 31 44 54 55 77 93
inserted 20


17 26 54 77 93 31 44 55 20
Need to insert 31 back into the sorted list
17 26 54 77 93 44 55 20
93>31 so shift it to the right
17 26 54 77 93 44 55 20
77>31 so shift it to the right
17 26 54 77 93 44 55 20
54>31 so shift it to the right
17 26 31 54 77 93 44 55 20
26<31 so insert 31 in this position

Visualizations

- Let’s take a look at some visualizations!