Insertion Sort

- Comparison-based sorting algorithm
- The lower part of the array (or sub-list) is always sorted prior to other elements
- The algorithm searches the array and move the unsorted elements into sorted array.

Why Insertion Sort?

- ✓ Efficient for sorting data that was already sorted
- ✓ Sort the array while receiving new data
- ✓ Requires small amount of memory to execute
- ✓ Does not require lot of code to use and does not change order of “like” elements in array

Example

Sort the given array

11	4	17	18	2	22	1	8

Algorithm

11	4	17	18	2	22	1	8

Step 1) Compare the adjacent value

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) If not, insert the value

Step 5) Continue Step 2 & 3 from 1 to (n-i)

Compare

11	4	17	18	2	22	1	8

Step 1) Compare the pivot and next

Compare

4	11	17	18	2	22	1	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Compare

4	11	17	18	2	22	1	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Compare

4	11	17	18	2	22	1	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Move

4	11	17	18	2	22	1	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Compare

4	11	17	18	2	22	1	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Move

4	11	17	18	2	22	1	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Move

4	11	17	18	2	22	1	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Move

4	11	17	2	18	22	1	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

4	11	17	2	18	22	1	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

4	11	17	2	18	22	1	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

4	11	2	17	18	22	1	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

4	11	2	17	18	22	1	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

4	2	11	17	18	22	1	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

4	2	11	17	18	22	1	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

2	4	11	17	18	22	1	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

2	4	11	17	18	22	1	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

2	4	11	17	18	22	1	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

2	4	11	17	18	22	1	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

2	4	11	17	18	1	22	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

2	4	11	17	18	1	22	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

2	4	11	17	1	18	22	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

2	4	11	17	1	18	22	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

2	4	11	1	17	18	22	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

2	4	11	1	17	18	22	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

2	4	1	11	17	18	22	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

2	4	1	11	17	18	22	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

2	1	4	11	17	18	22	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

2	1	4	11	17	18	22	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

1	2	4	11	17	18	22	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot

Check

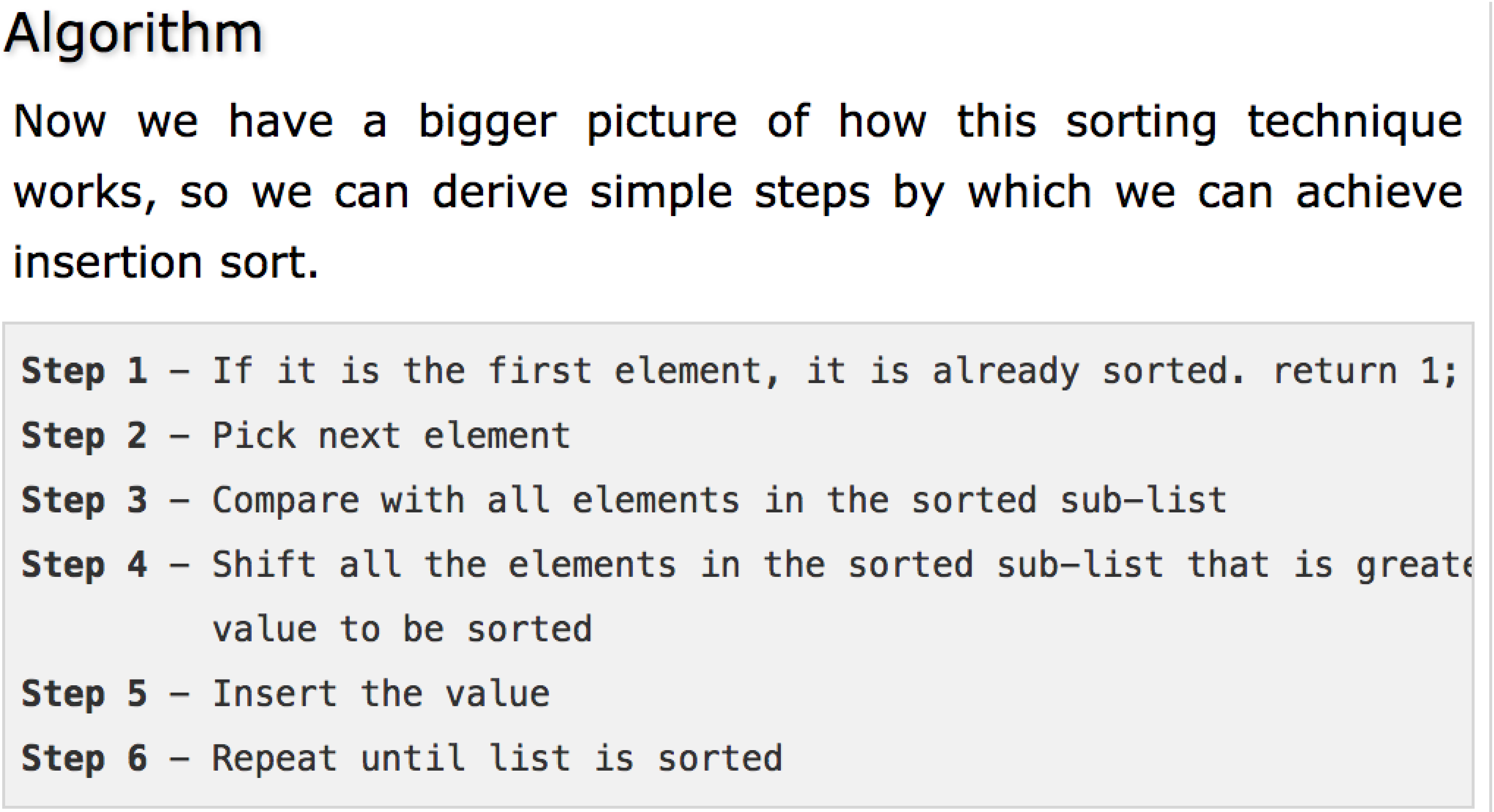
1	2	4	11	17	18	22	8

Step 1) Compare the pivot and next

Step 2) If the left index is greater than the right index, then swap

Step 3) Also it checks all the elements in the left for sorted order

Step 4) Yes, move the pivot



Algorithm

Now we have a bigger picture of how this sorting technique works, so we can derive simple steps by which we can achieve insertion sort.

```
Step 1 − If it is the first element, it is already sorted. return 1;
Step 2 − Pick next element
Step 3 − Compare with all elements in the sorted sub-list
Step 4 − Shift all the elements in the sorted sub-list that is greate
         value to be sorted
Step 5 − Insert the value
Step 6 − Repeat until list is sorted
```