

## Sorting-

Arranging data in a particular order

Algorithms in sorting?

Specifies the way to arrange data in a particular order

Why is sorting important?

- Data searching can be optimized to a very high level, if data is stored in sorted manner.
- Telephone Directory
- Dictionary

In-place sorting and Not-in-place sorting

- Extra space for storage of elements while performing sorting; and that space is equal to  $\Theta(n)$  greater than the numbers of elements being sorted; called as not-in-place sorting.  $\langle$  Merge sort  $\rangle$
- in-place sorting  $\langle$  Bubble sort  $\rangle$

Stable and not-stable sorting-

- sorting algorithm which doesn't change the sequence of similar content in which they appear, it is called stable sorting

35, 33, 42, 10, 14, 19, 26, 44, 26, 31

10, 14, 19, 26, 26, 31, 33, 35, 42, 44

- if it changes the sequence of similar content in which they appear, it is called as unstable sorting.

35, 33, 42, 10, 14, 19, 26, 44, 26, 31

10, 14, 19, 26, 26, 31, 33, 35, 42, 44

Matters when we wish to maintain the sequence of original elements.

Adaptive and non-adaptive sorting algorithm -

- adaptive when it takes advantage of already sorted elements in the list that is to be sorted... Don't swap them.

- Non-adaptive are opposite

Increasing order -

order of data if the successive element is greater than the previous one.

1, 3, 4, 6, 8, 9

Decreasing order -

9, 8, 6, 4, 3, 1

A sequence of values are in decreasing order if the successive element is less than the current one.

9, 8, 6, 4, 3, 1

Non-decreasing order-

successive element is greater than or equal to the current element.

1, 3, 3, 6, 8, 9

Non-increasing order-

successive element is less than or equal to the current element

9, 8, 6, 3, 3, 1

Bubble sort-

- Simple sorting algorithm
- Comparison based algorithm
- pair of adjacent elements are compared
- and elements are swapped if they are not in order.

$$T \approx O(n^2) \quad S \approx O(1)$$

When we have large datasets, this would be bad.

Algorithm-

14      33      27      35      10

14      33      27      35      10  
Swap

14      27      33      35      10

14      27      33      35      10  
Swap

14      27      33      10      35

Bigger element  
at the  
end

14      27      33      10      35

14      27      33      10      35

14      27      33      10      35  
Swap

14      27      10      33      35

14      27      10      33      35

14      27      10      33      35

14      27      10      33      35  
Swap

14      10      27      33      35

14      10      27      33      35  
Swap

10 14 27 33 35

```
public void bubbleSort (int [] arr) {  
    for (int i=0; i< arr.length-1; i+=1) {  
        boolean swapped = false;  
        for (int j=0; j< arr.length-1; j+=1) {  
            if (arr[i] > arr[i+1]) {  
                swap (arr, i, i+1);  
                swapped = true;  
            }  
        }  
        if (!swapped) {  
            break;  
        }  
    }  
}
```

$$T = O(n^2)$$
$$S = O(1)$$