**Bubble sort :**

- it is in-place algorithm (we are not making new array)

- O(n^2) time complexity – quadratic

100 steps to sort 10 items

10 000 to sort 100 items …

An array is partitioned logically for sorted and unsorted partition

unsortedPartitionIndex – last index of array, because whole array isn't sorted

i = 0 – index used to traverse array from left to right

Operations:

_____

dn – do nothing

swp – swap elements

_____

i = 0

unsortedPartitionIndex = 6 (Array size)

| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | |
|---|---|---|---|---|---|---|---|---|
| Values | 20 | 35 | -15 | 7 | 55 | 1 | -22 | Decsription |
| STEP ( i ) | | | | | | | | |
| 0 | 20 | 35 | -15 | 7 | 55 | 1 | -22 | 20 < 35, **dn** |
| 1 | 20 | -15 | 35 | 7 | 55 | 1 | -22 | 35 > - 15, **swp** |
| 2 | 20 | -15 | 7 | 35 | 55 | 1 | -22 | 35 > 7, **swp** |
| 3 | 20 | -15 | 7 | 35 | 55 | 1 | -22 | 35 < 55, **dn** |
| 4 | 20 | -15 | 7 | 35 | 1 | 55 | -22 | 55 > 1, **swp** |
| 5 | 20 | -15 | 7 | 35 | 1 | -22 | 55 | 55 > -22, **swp** |

55 is on right position, so now we decrease unsortedPartitionIndex by one

We are repeatingevery step till i > 0

Bubble sort can be realized by:

Moving elements from start to end of array

Moving elements from end to start of array