* Bubble Sort *

→ ith round → ith largest element place at right place

→ In each round largest element will place at it's position

arr[]: | 10 | 1 | 7 | 6 | 14 | 9 |

swap. (a > b) → swap (10 > 1) → swap.

```
for(i=1; i<n; i++){
    for(j=0; j<n-i; j++){
        if(arr[j] > arr[j+1]){
            swap( );
        }
    }
}
```

Round 1:

1 10 7 6 14 9 — swap

1 7 10 6 14 9 — swap

1 7 6 10 14 9 — ignore

1 7 6 10 14 9 — swap

1 7 6 10 9 | 14 — sorted

Round 2:

1 7 6 10 9 | 14 — ignore, sorted

1 7 6 10 9 | 14 — swap, sorted

1 6 7 10 9 14 — ignore

1 6 7 10 9 14 — swap

1 6 7 9 | 10 14 — sorted

→ like wise Round 3, 4, 5 will be done

Time complexity: $O(n^2)$ Space Complexity: constant

best case: $O(n)$

→ It is stable Algo: because we swap only when $a > b$. So relative order of equal element will be maintained.

→ In place sorting: The sorting Algo that sorts the input array inplace without using any addition memory.