⓪ INSERTION SORT

is the sorting mechanism where the sorted array is built having one item at a time.

These array elements are compared with each other SEQUENTIALLY and then arranged SIMULTANEOUSLY in some particular order.

This sort works on PRINCIPLE OF INSERTING AN ELEMENT AT A PARTICULAR POSITION, Hence the name is INSERTION SORT.

| 5 | 1 | 6 | 2 | 4 | 3 |

5 ① 6 2 4 3

1 5 ⑥ 2 4 3

1 5 6 ② 4 3

1 2 5 6 ④ 3

1 2 4 5 6 ③

| 1 | 2 | 3 | 4 | 5 | 6 |

Insertion Sort : Ek comparison based sorting technique hai, jo array ko sort krti hai by shifting the element one-by-one from an unsorted array (sub) to sorted (sub) array.

* NOTE TO REMEMBER *

Array of Size = 1 is always sorted.

| 5 | 1 | 6 | 2 | 4 | 3 |

ITERATION 1

1.1 1 5̶ 1̶5 6 2 4 3

1.2 1 5 6 2 4 3

∴ 1 5 6 2 4 3

ITERATION 2

2.1 1 5 2̶6 2̶6 4 3

2.2 1 2̶5 2̶5 6 4 3

2.3 1 2 5 6 4 3

∴ 1 2 5 6 4 3

ITERATION 3

3.1 1 2 5 4̶6 4̶6 3

3.2 1 2 4̶5 5̶4 6 3

3.3 1 2 4 5 6 3

∴ 1 2 4 5 6 3

ITERATION 4

4.1 1 2 4 5 3̶6 3̶6

4.2 1 2 4 3̶5 5̶3 6

4.3 1 2 3̶4 3̶4 5 6

4.4 1 2 3 4 5 6
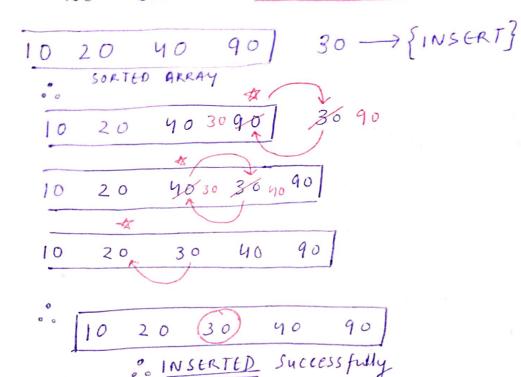
∴ 1 2 3 4 5 6 SORTED

● INSERTION SORT me REVERSE BUBBLINH Technique use
hati hai !

```java
public static void main (String [] args) {
Scanner s = new Scanner (System.in);
int n = s.nextInt();
int [] arr = new int [n];
for (int i=0; i< arr.length; i++){
    arr[i] = s.nextInt();
}

for (int i=1; i<=arr.length-1; i++)      → (n-1) tak journey hogi!
{
    for (int j= i-1; j>=0; j--)          → (i-1) se (0) tak sab elements
    {                                       compare hongey !
        if (arr[j] > arr[j+1])           → Agar (j+1) wala element
        {                                    bada hua (j) walay element
            int temp = arr[j];               se toh swaPPINh Hogi !
            arr[j] = arr[j+1];
            arr[j+1] = temp;
        } else {
            break;                       → end me break
        }                                   lagega!
    }
}
for (int val : arr){
    System.out.print (val + " ");
}
System.out.println (".");
}
}
```

if we want to add element in our sorted array then, we will use **REVERSE BUBBLING** technique.

| 10 | 20 | 40 | 90 | 30 → {INSERT}
SORTED ARRAY

∴

| 10 | 20 | 40 30 90 | 30 90

| 10 | 20 | 40 30 30 40 90 |

| 10 | 20 | 30 | 40 | 90 |

∴

| 10 | 20 | (30) | 40 | 90 |
∴ INSERTED Successfully

Time complexity:

We used ② loops.

∴ For each (n-1) iteration

∴ we will perform

| 1 + 2 + 3 + ...n swaps |

WORST CASE? → when all the numbers are sorted in reverse order.

$\textcircled{A.P}$

∴ $\dfrac{n(n+1)}{2}$ ←

↓

$\textcircled{n^2}$ ∴

worst CASE : $O(n^2)$

Best Case : $O(n)$

Space complexity

$O(1)$