SORTING – Selection Sort, Bubble Sort, Insertion Sort

(I) Selection Sort

13	46	24	52	20	9
0	1	2	3	4	5

ascending order → 9, 13, 20, 24, 46, 52

9 46 24 52 20 13 Step 1 (not sorted)

13 24 52 20 46 Step 2

20 ~~52~~ 24 46 Step 3

24 52 46 Step 4

24 46 52 Step 5

9 13 20 24 46 52

"Get the minimum element and swap it with the first element of the unsorted part"

Pseudo Code:-

swap happened at 0 & min. index {0 to n-1}

" " " 1 & " " {1 to n-1}

" " " 2 & " " {2 to n-1}

⋮ ⋮

n-2

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

0 to N
1 to N-1

OR swap(arr[j], arr[i])

$N + (N-1) + (N-2) + \dots + 2$

$= \frac{N(N+1)}{2}$

$= \frac{N^2}{2} + \frac{N}{2}$

$\approx O(N^2)$

Best

worst T.C

avg

11 Bubble Sort → pushes the max. to the last by adjacent swaps.

13, 46, 24, 50, 20 (9)
13, 24, 46, 50, 20, 9
13, 24, 46, 20, 50, 9
13, 24, 46, 20, 9, 50
} 1 round of adjacent swap (1) 0 to n-1

13, 24, 20, 46, 9, 50
13, 24, 20, 9, 46, 50
} round 2 0 to n-2

13, 20, 24, 9, 46, 50
13, 20, 9, 24, 46, 50
} round 3 0 to n-3

13, 20, 9
} round 4 0 to n-4

13, 9, 20
} round 5 0 to n-5

9, 13, 20, 24, 46, 50

T.C
n runs
n-1
n-2
n-3

Pseudo Code:-

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

0 to n-2
0 to n-3
0 to n-n

$T.C = n \times \frac{(n+1)}{2} \approx O(n^2)$ Worst/Avg complexity

Interview Quest

$T.C = O(n)$ Best case
↳ i.e if array was sorted only 1, 3, 5, 9, 12 then after checking all adjacent in round 1, no swap was done, thus after that the checking stops. ⇒ will get linear T.C O(N) because array was sorted & would break out.

III) INSERTION SORT → Takes an element and places it in correct order.

14, 9, 15, 12, 6, 8, 13

9, 14, 15, 12, 6, 8, 13

9, 12, 14, 15, 6, 8, 13

6, 9, 12, 14, 15, 8, 13

6, 8, 9, 12, 14, 15, 13

6, 8, 9, 12, 13, 14, 15

Pseudo Code:-

```
for (int i=0; i<n; i++) {
    ~~for (int~~ j=i;
    while (j>0 && a[j-1] > a[j]) {
        swap (a[j-1], a[j]);
        j--;
    }
}
```

T.C

0
+1
+2
+3
⋮
+n

$= \frac{n(n+1)}{2}$

$\approx O(n^2)$ worst & avg case

T.C = O(n) best case

1 2 3 4 5