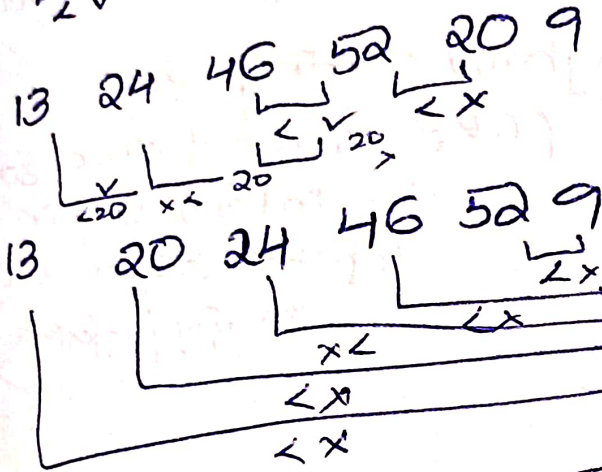
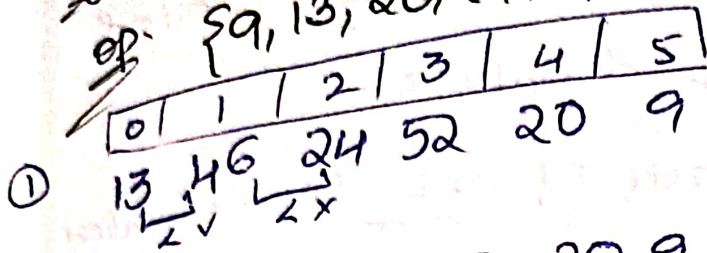


③ Insertion Sort: [Partial Sorting] ← Sorting by parts

$N=6$, arr[] = {13, 46, 24, 52, 20, 9}

eg: {9, 13, 20, 24, 46, 52}



{9, 13, 20, 24, 46, 52}

Algo: Left Hand Side of Array gets Sorted

```
for(int i=0; i<=n-2; i++) {
    for(int j=i+1; j>0; j--) {
        if(a[j]<a[j-1]) {
            swap(a[j], a[j-1]);
        }
        else {
            break; // if j>j-1
        }
    }
}
```

Best case

TC: $O(N)$

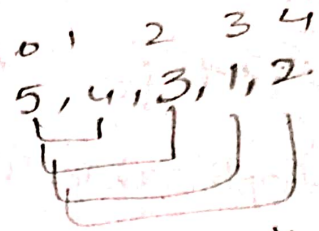
If arr is already sorted

Worst

TC: $O(N^2)$

SC: $O(1)$

Also stable sorting.



9 5 3 1 2 sort until 1st index
3 4 5 1 2 sort until 2nd index
1 3 4 5 2
1 2 3 4 5 ✓
3rd
4th.

i j > 0
1st pass 0 1
2nd pass 1 2
3rd pass 2 3
4th pass 3 4
(0 to n-2)

j
5 4 3 1 2
j < j-1 (swap)
4 5 3 1 2
j > 0

j = i+1, j > 0; j--

Why Insertion Sort:

- 1) Stable sort (order of ele maintained)
- 2) Adaptive (breaks loop, when arr is sorted)
- 3) Used for smaller 'n' values
- 4) Applicable when parts of arr are sorted
- 5) Takes part in hybrid sorting algo.
(like merge & quicksort are internally in default sort method in py / java)