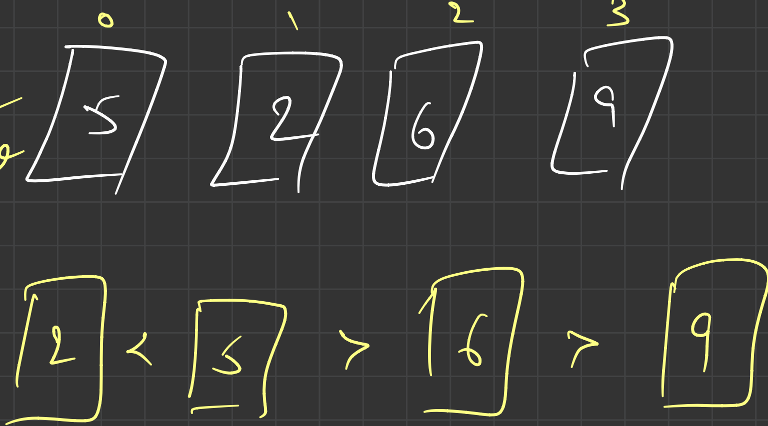
Insertion Sort

we dont swap we shift elements

→ like Card Game
(1-1 karke Card diye jayege usko sorted order me lagane hai)

Assume that it is at right place ←



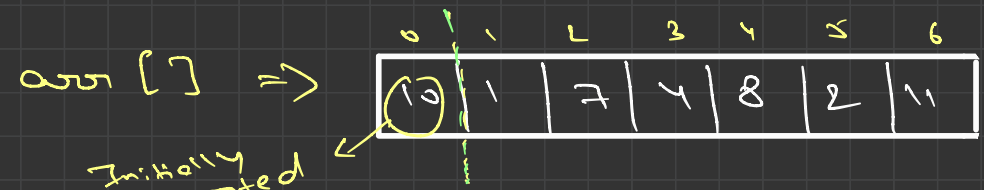
---

4 12 11 20

4 12 (12 > 4)

4 11 12 (11 < 12, 11 > 4)

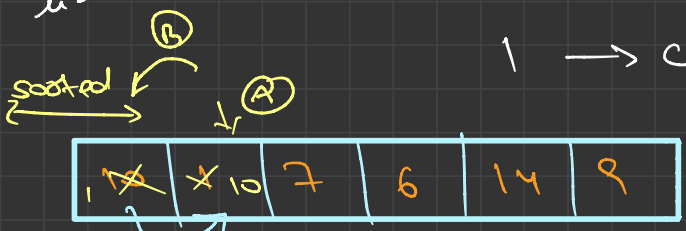
4 11 12 20 (20 > 12)



Round 1 => 1 < 10 → left side

i = 1 10 → index 1st

1 → copy to 0th index



① → value = 1 fetch

② → 1 < 10 comparison

③ → shift → 10 shifting

④ → empty space → insert 1 copy

Round 2 => 1 10 7 4 8 2 11

i = 2

7 < 10 → left

7 > 1 → right

Round 3 =>

i = 3

1 7 10 4 8 2 11

4 inside

4 < 10 (left)

4 < 7 (left)

4 > 1 (right)

Round 4 ⇒ 1 4 7 10 8 2 11

i = 4

8 < 10 → left

8 > 10 → right

Round 5

i = 5

1 4 7 8 10 2 11

2 < 10

2 < 8

2 < 7

2 < 4 } left

2 > 1 → right

Round 6

i = 6

1 2 4 7 8 10 11

11 > 10 → right

All array are sorted

Space complexity = O(1)

Time complexity = $O(n^2)$

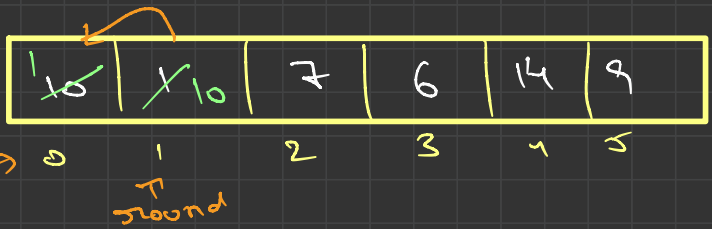
Best case = O(n)

because 0 index ko sort nhi kar skta to assume sorted

```
for (int pround = 1 ; pround < n ; pround++)
{
    // Step 1 = fetch
    temp = arr[pround]

    //Step 2 = compare
    int j = pround-1;
    for ( ; j >= 0 ; j--)
    {
        if ( arr[j] > temp )
        {
            // Step 3 = Shift
            arr[j+1] = arr[j]
        }
        else {
            break ; }
    }
    arr[j+1] = temp ; //copy
}
```

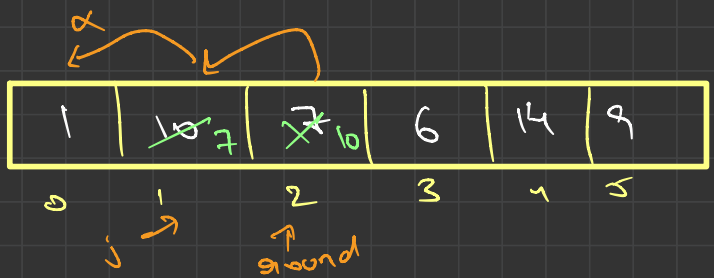
bug we have declare outside the j loop.



Round 1

vol = 1 10 > 1

↳ shifting



Round 2

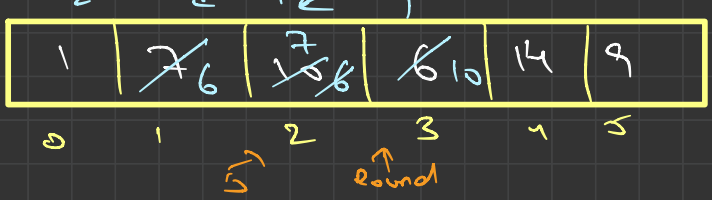
vol = 7 10 > 7

↳ shifting

1 > 7

↳ break

arr[j+1] = vol



Round 3

vol = 6

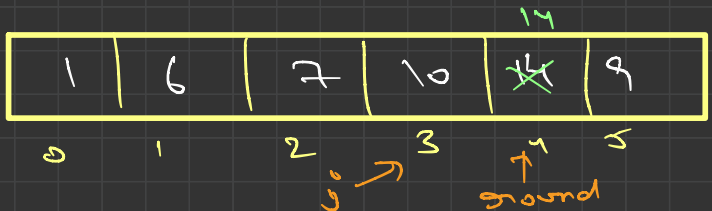
10 > 6 ⇒ shift

7 > 6 ⇒ shift

1 > 6 ⇒ break

arr[j+1] = vol;

Round 4

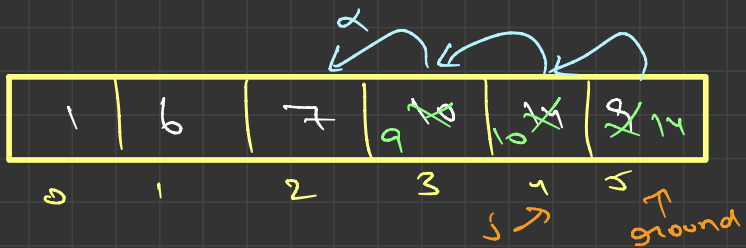


val = 14

10 > 14 → break

arr[j+1] = val;

Round 5



val = 9

14 > 9 → shift

10 > 9 → shift

7 > 9 → break;

arr[j+1] = val;

Final Result