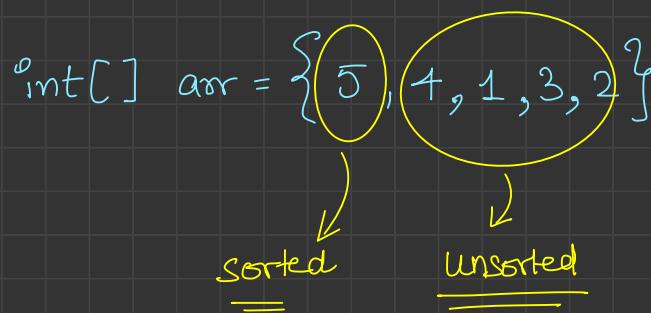


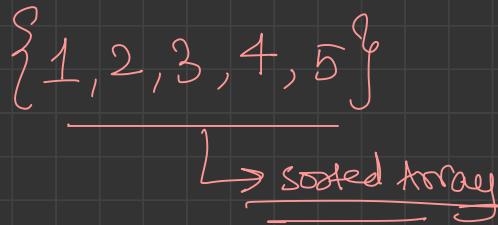
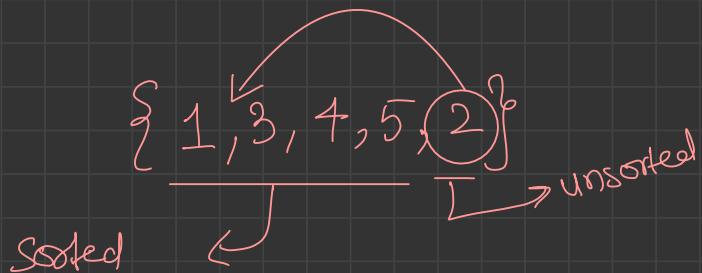
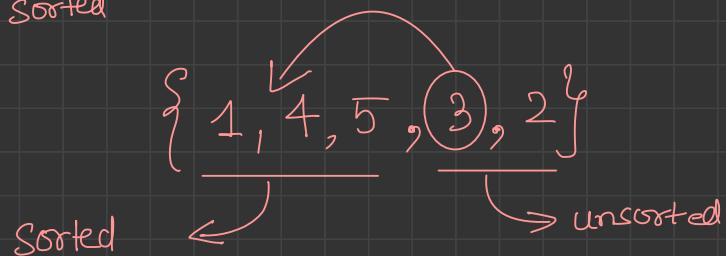
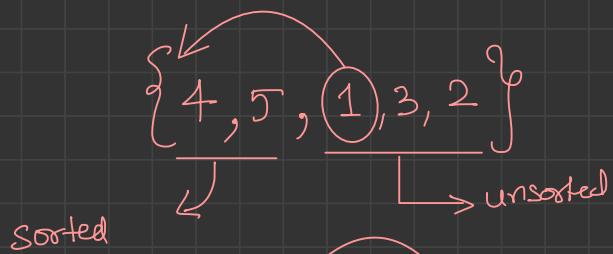
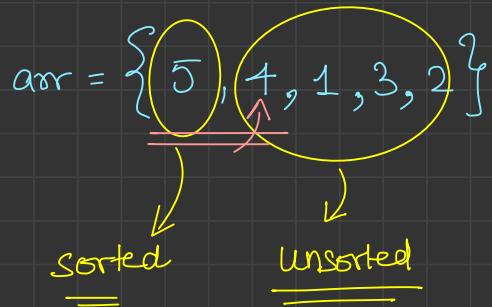


## Insertion Sort

→ Inspiration: Come from way of sorting cards we have in hand.

- \* select a element and place it at its correct position.





Q  
Insertion

$\{ [1, 3, 4, 5], 2 \}$

Swap them

$\{ 1, 2, 3, 4, 5 \}$

↑

$\{ 1, 3, 4, 2, 5 \}$

Swap them

But  $i = N - 1$  ;  
 while ( $i >= 1$  &  $\{ arr[i-1] > arr[i] \}$ )

$\{ 1, 3, 2, 4, 5 \}$

Swap them

$\{$   
 Swap( $arr[i]$ ,  $arr[i-1]$ );  
 $i = i - 1$ ,  
 $\}$

$\{ 1, 2, 3, 4, 5 \}$

→  $\boxed{\underline{\underline{\text{break}}}} \mid \underline{\underline{\text{stop}}}$

- $\text{arr} = \{5, 4, 1, 3, 2\}$  insertion (0,0)
- $\{4, 5, 1, 3, 2\}$  insertion (0,1)
- $\{4, 1, 5, 3, 2\}$
- $\{1, 4, 5, 3, 2\}$  insertion (0,2)
- $\{1, 4, 3, 5, 2\}$
- $\{1, 3, 4, 5, 2\}$  insertion (0,3)
- $\{1, 3, 4, 2, 5\}$
- $\{3, 2, 1, 4, 5\}$
- $\{1, 2, 3, 4, 5\}$
- Sorted Array!

for (int  $i = 1$ ;  $i < n$ ;  $i++$ )

{

int  $j = i$ ;

while ( $j \geq 1 \&& arr[j-1] > arr[j]$ )

{

swap( $arr[j-1], arr[j]$ );

$j--$

}

}

$\{$  0 1 2 3 4  
 1, 2, 3, 4, 5  $\}$



$\checkmark$   
 $\{$  TC:  $O(N^2)$       SC:  $O(1)$   $\}$

for ( $i = 1$ ;  $i < n$ ;  $i++$ )  
 {  
      $int j = i$ ;  
     while ( $j >= 1$  &&  $arr[j-1] > arr[j]$ )

$\{$       swap( $arr[j-1], arr[j]$ );  
      $j--$ ;  $\checkmark$   
 $\}$

Sorted Array

$\{ 1, 2, 3, 4, 5 \}$

$Tc : O(n)$

$Sc : O(1)$

NOTE : Worst and avg Time Complexity of Insertion Sort is  $O(n^2)$ ,

whereas best case is  $O(n)$  } for unsorted array

# Sort 0-1 Array

int [] arr = { 0, 1, 2, 3, 4, 5, 6, 7, 8 }  
                { 1, 1, 0, 0, 0, 1, 0, 1, 0 }

↓ sort

{ 0, 0, 0, 0, 0, 1, 1, 1, 1 } ✓

Brute force .

① Arrays.sort(arr);

Tc: O(NlogN), Sc: O(1)

Better?

$$\begin{array}{r} 0 \ 0 \ 0 \ 0 \ 0 \\ \hline 0 \ 1 \ 2 \ 3 \ 4 \end{array} \quad \begin{array}{r} 1 \ 1 \ 1 \ 1 \\ \hline 5 \ 6 \ 7 \ 8 \end{array}$$

$\text{int[]} \text{ arr} = \{ 1, 1, 0, 0, 0, 1, 0, 1, 0 \}$

$\downarrow \text{sort}$

$\{ 0, 0, 0, 0, 0, 1, 1, 1, 1 \}$

$\text{TC} : O(2^N)$   
 $\text{SC} : O(1)$

$\left\{ \begin{array}{l} \text{cnt } 0 = 1 \neq 2 \neq 5 \\ \text{cnt } 1 = 4 \neq 3 \neq 4 \end{array} \right.$

$i \rightarrow 0 \rightarrow (\text{cnt } 0 - 1)$   
 $i \rightarrow \text{cnt } 0 \rightarrow (N - 1)$

$\left\{ \begin{array}{l} \text{Can you do in one pass} \\ \text{only} \\ \text{strictly } O(n) \end{array} \right.$

# Best Approach

int]] arr = { 0 1 2 3 4 5 6 7 8  
              |-----|-----|  
              \|    \|  
              hi   lo }

0 → lo - 1 → 0  
hi + 1 → n - 1 → 1

while (lo <= hi) {  
    if (arr[lo] == 1)  
        {  
            swap(arr[lo], arr[hi]);  
            hi--;  
        }  
    else {  
        lo++;  
    }  
}

$\text{int arr} = \{ 0, 1, 2, 3, 4, 5, 6 \}$

    $\uparrow$   $\uparrow$   $\uparrow$   
 $\text{lo} \quad \text{hi}$

$\text{lo} = 0$   
 $\text{hi} = n - 1$

while ( $\text{lo} < \text{hi}$ ) {  
    if ( $\text{arr}[\text{lo}] == 1$ ) {  
         $\checkmark \text{swap}(\text{arr}[\text{lo}], \text{arr}[\text{len}])$ ,  
         $\text{hi}--$   
    }  
    else {  
         $\text{lo}++$   
    }  
}

$\boxed{\text{TC}: O(N)}$   
 $\boxed{\text{SC}: O(1)}$

# Sort 0-2-1 Array

$$\text{int arr} = \{ 2, 2, 1, 0, 1, 0, 2, 1, 1, 0 \}$$

↓  
sort

$$\stackrel{\text{O}(P)}{=} \{ 0, 0, 0, 2, 2, 2, 1, 1, 1, 1 \}$$

Brute force

→ Cut one of each Ele, and then add as per one.

$$\boxed{TC: O(2N) \ SC: O(1)} =$$

Better Approach

$\text{int arr} = \{2, 2, 1, 0, 1, 0, 2, 1, 1, 0\}$

lo      mid      hi

$$\begin{array}{l} 0 \rightarrow \text{lo} - 1 \\ \hline \text{lo} \rightarrow \text{mid} - 1 \\ \hline \text{hi} + 1 \rightarrow N - 1 \end{array} \longrightarrow \begin{array}{l} 0's \\ 2's \\ 1's \end{array}$$

$\text{mid} \rightarrow \text{hi}$        $\longrightarrow$  unknown array

$\{$   $0, 0, 0, 0, 0$ ,  $2, 2, 2, 2$ ,  $0, 1, 2, 2, 1, 0, 1, 0$ ,  $1, 1, 1, 1, 1$   $\}$ 
  
 ⌈
   
 ⌊  $l_0$        $mid$        $hi$  ⌋

$N-1$

$0 \rightarrow l_0 - 1 \rightarrow 0's \checkmark$   
 $\checkmark l_0 \rightarrow mid - 1 \rightarrow 2's \checkmark$   
 $hi + 1 \rightarrow N-1 \rightarrow 1's \checkmark$

$mid \rightarrow hi \rightarrow \underline{\text{unknown array}} \checkmark$

$\text{int arr} = \{0, 0, 0, 2, 2, 2, 1, 1, 1, 1\}$

hi  
mid  
lo

while (mid < hi) {

if (arr[mid] == 1)  
{ swap(arr[mid], arr[hi]);  
hi--; }

else if (arr[mid] == -2)

{ mid++; }

else  
{ swap(arr[lo], arr[mid]); mid++; lo++; }

```

static int[] Sorting021(int n,int[] arr){
    // Write your code here
    int lo = 0;
    int mid = 0;
    int hi = n - 1;

    while (mid <= hi) {
        if (arr[mid] == 2) ✓
            mid++;
        } else if (arr[mid] == 1) ✓
            int temp = arr[hi];
            arr[hi] = arr[mid];
            arr[mid] = temp;

            hi--;
        } else { ✓
            int temp = arr[lo]; } }

            mid++;
            lo++;
        }

    return arr;
}

```

$arr[] = \{0, 0, 0, 0, 2, 2, 1, 1\}$   
 lo ↑      hi ↑  
 ↑ mid  
 (mid → hi) → undefined.

$$\left\{ \begin{array}{l} TC: O(N) \\ SC: O(1) \end{array} \right.$$

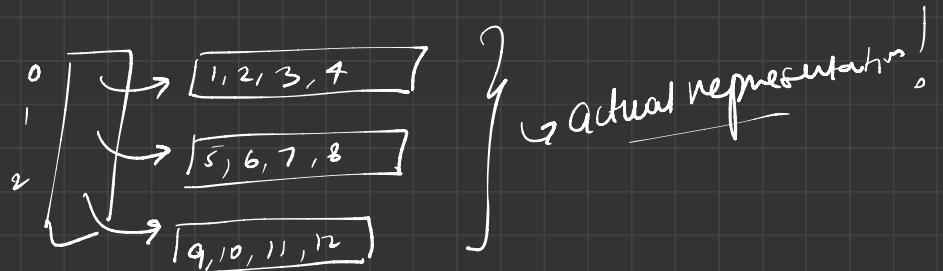
AS Sorting 1

asc. order!

$$\text{int} [] [] \text{mat} = \begin{bmatrix} c_0 & c_1 & c_2 \\ 10 & 11 & 9 \\ 1 & 2 & 10 \\ 17 & 15 & 2 \\ 16 & 19 & 13 \end{bmatrix}$$

$N \times M$   
 $4 \times 3$

$$\text{int}[ ] \text{ mat} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$



for (int i = 0; i < r; i++)

{  
    int [ ] arr = mat [i];

}

$$\begin{matrix} & 0 & 1 & 2 \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} & \left[ \begin{matrix} 10 & 11 & 9 \\ 4 & 2 & 10 \\ 17 & 15 & 2 \\ 16 & 19 & 13 \end{matrix} \right] & N \times M \\ & 4 \times 3 \end{matrix}$$

Transpose  $\Rightarrow$

$$\left[ \begin{matrix} 10 & 4 & 17 & 16 \\ 11 & 2 & 15 & 14 \\ 9 & 10 & 2 & 13 \end{matrix} \right] M \times N$$

$\downarrow$  all columns are sorted!

$\downarrow$  Sort Each Row

$$\left[ \begin{matrix} 9 & 2 & 2 \\ 10 & 11 & 9 \\ 16 & 19 & 10 \\ 17 & 15 & 13 \end{matrix} \right] 4 \times 3$$

Transpose

$$\left[ \begin{matrix} 9 & 10 & 16 & 17 \\ 2 & 11 & 14 & 15 \\ 2 & 9 & 10 & 13 \end{matrix} \right]$$