



# Sorting

↳ `Arrays.sort()` { Sort the array in asc. order }

int[] arr = { 5, 3, 2, 7, 1 };

`Arrays.sort(arr);`  
↳ { 1, 2, 3, 5, 7 } ✓

{ TC:  $O(N \log N)$  ↳ size of the array  
SC:  $O(1)$

→ Bubble Sort

int C arr = {  
0 1 2 3 4  
5, 3, 4, 1, 2}

{ 3, 5, 4, 1, 2 }

{ 3, 4, 5, 1, 2 }

{ 3, 4, 1, 5, 2 }

{ 3, 4, 1, 2, 5 }

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

of unsorted array  
Bigest ele h at the end.

## Selection Sort

int [] arr = { 5, 4, 2, 1, 3 }

→ Select smallest ele. and  
Place first pos. of unsorted part.

{ 1, 4, 2, 5, 3 }



TC:  $O(n^2)$   
SC:  $O(1)$

## Insertion Sort

int [] arr = {5, 4, 2, 3, 1}  $\xrightarrow{\text{TC: } O(N^2)}$

{1, 2, 3, 4, 5}

TC: Best case  $O(N)$   
SC:  $O(1)$

{4, 5, 2, 3, 1}  $\xrightarrow{\text{TC: } O(N^2)}$

{2, 4, 5, 3, 1}  $\xrightarrow{\text{TC: } O(N^2)}$

{2, 3, 4, 5, 1}  $\xrightarrow{\text{TC: } O(N^2)}$

Avg Case OR Worst Case

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

{1, 2, 3, 4, 5} ✓ Sorted Array

## Insertion Sort

int [] arr = {5, 4, 2, 1, 3}

Sorted Array

{ 1, 2, 4, 3, 5 }

{ 4, 5, 2, 1, 3 }

{ 1, 2, 3, 4, 5 }

{ 4, 2, 5, 1, 3 }

{ 2, 4, 5, 1, 3 }

{ 2, 4, 1, 3 }

{ 2, 1, 4, 5, 3 }

{ 1, 2, 4, 5 }



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

int  $j = i$ ;

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

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

$i--$ ;

}

}

arr[] = {1, 2, 3, 4, 5}



Sorted Array

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

AS Sort 5  
0

int[] arr = { 0, 1, 2, 3, 4, 5, 7, 4, 1, 5, 3, 2 }  
As smaller as possible

w, x, y, z

$$[arr[w] * arr[x] - [arr[y] * arr[z]]]$$

As bigger as possible      Maximized

$$7 * 5 - 2 * 1 = \boxed{33}$$

✓

$\left\{ \begin{array}{l} \text{arr}[w] \rightarrow \text{max value} \\ \text{arr}[x] \rightarrow \text{second max} \\ \text{arr}[y] \rightarrow \text{second min} \\ \text{arr}[z] \rightarrow \text{min value} \end{array} \right.$

$\text{int}[] \text{ arr} = \{ \begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 7 & 4 & 1 & 5 & 3 & 2 \end{matrix} \}$

Arrays, Sort (arr) ;  $\rightarrow$   $TC: O(N \log N)$   
 $SC: O(1)$

$\{ 1, 2, 3, 4, 5, 7 \}$

$$(arr[n-1] * arr[n-2] - arr[1] * arr[0])$$

$\downarrow \qquad \downarrow \qquad \downarrow \qquad \downarrow \qquad \downarrow$

max1      max2      min1      min2      ans

`int[] arr = { 0, 1, 2, 3, 4, 5 }  
 ↗ ↗ ↗ ↗ ↗ ↗ }`

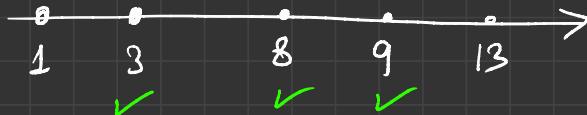
`min1 = Integer. MIN_VALUE; } TC: O(N)`

`max2 = Integer. MAX_VALUE; } TC: O(N)`

`{ TC: O(4N) ~ O(8N) } SC: O(1)`



AS Sorting 4.



$$l = 6$$

$$k =$$

$$(1, 3, 8) \rightarrow 8 - 1 = 7 \quad X$$

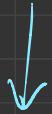
$$(1, 3, 9) \rightarrow 9 - 1 = 8 \quad X$$

$$(1, 3, 13) \rightarrow 13 - 1 = 12 \quad X$$

$$(1, 8, 9) \rightarrow 9 - 1 = 8 \quad X$$

$$(1, 8, 13) \rightarrow 13 - 1 = 12 \quad X$$

$$\checkmark (3, 8, 9) \rightarrow 9 - 3 = 6 \quad \checkmark$$



$$\{9, 1, 3, 8, 13\}$$

↓ sort

$$\{1, 3, 8, 9, 13\}$$

i    j    k

for (i → 0 → n)

  for (j → i + 1 → n)

    for (k → j + 1 → n)