Sorting

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

sort in ascending order

{ 1, 2, 3, 4, 5 }

Sorting

1. Bubble Sort Algorithm
2. Selection Sort
3. Insertion Sort
4. Merge Sort
5. Quick Sort

(1–3) → learn here

(4–5) → some further Module

Bubble Sort.

int[ ] arr = new int[5]

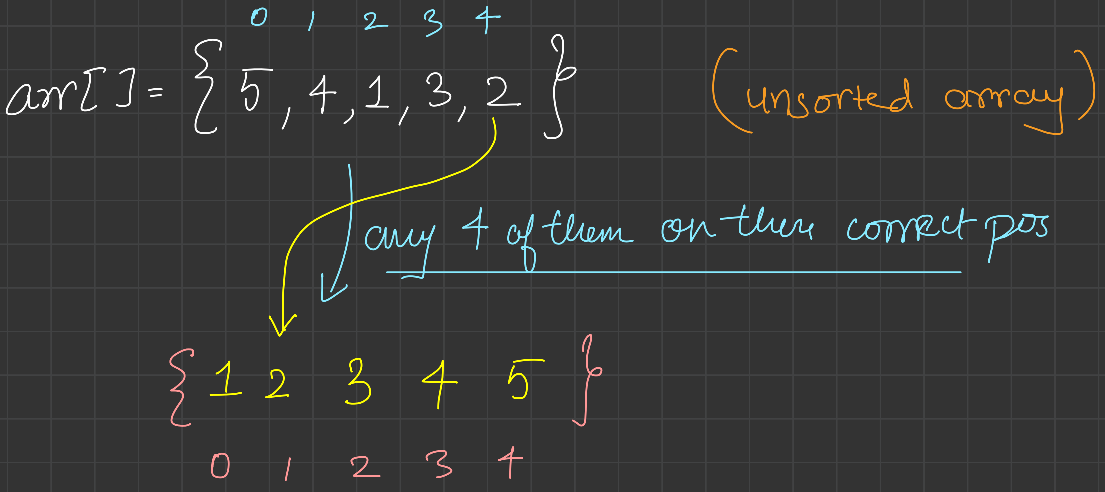
arr $\longrightarrow$ { 2, 5, 3, 1, 4 }

BubbleSort (arr)

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

what?
↳ Sort the elements in a ascending order

Bubble Sort -


0 1 2 3 4
arr[ ] = { 5, 4, 1, 3, 2 }
(unsorted array)
any 4 of them on their correct pos
{ 1 2 3 4 5 }
0 1 2 3 4

{ **NOTE:** if we have N elements in a unsorted array, By placing (N-1) elements on correct pos, we will get a sorted array.

Bubble Sort.

$$\text{int[] arr} = \left\{\begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ 4, & 5, & 2, & 3, & 1 \end{array}\right\} \longrightarrow \{1, 2, 3, 4, 5\}$$

$\Big\{ \longrightarrow$ I'll try to place the largest of the unsorted array at the last index of the unsorted array.

$$\boxed{\left\{\begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ \underbrace{1}_{\text{unsorted region}}, & \underbrace{2, & 3, & 4, & 5}_{\text{sorted region}} \end{array}\right\}} \longrightarrow \underline{\text{whole array is sorted}}.$$

Swap two elements

```
int a = 10;
int b = 20;

print(a + "   " + b);      ⟶  10   20

                 10
int temp = a;

20 ⟵ a = b;

10 ⟵ b = temp;

print(a + "   " + b)       ⟶  20   10
```

Swaping algorithm

0

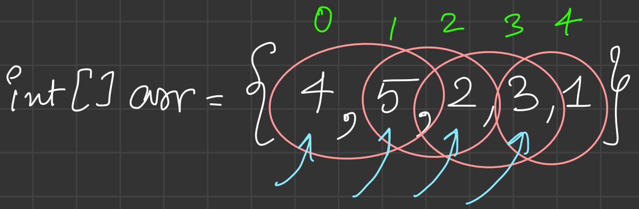
$$\begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ \end{array}$$

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

(1) i have to sort (N-1) Elements.

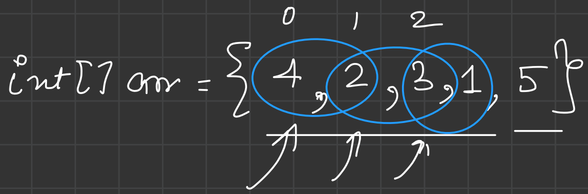
```
for (int i=1 ; i<=N-1; i++)
{
        work of sorting
}
```

$i = 1$



$\underline{N = 5 - i - 1}$

$i = 2$



$5 - 2 - 1 = \boxed{2}$

```
for (int i=1; i<=N-1; i++)
{
    for (int j=0; j<=N-i-1; j++)
    {
        if (arr[j] > arr[j+1])
            swap them
    }
}
```

$\rightsquigarrow$ { starting index of each Bubble }

$$\begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ \end{array}$$

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

if arr size N

No. of iterations $= (N-1)$

✓ 1st Ele at the End.

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

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

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

4th Iteration $\{\boxed{4, 2, 3, 1}\, \boxed{5}\}$ ✓

$\downarrow$ unsorted $\quad \downarrow$ sorted

✓ 2nd Ele at the End.

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

1st Iteration $\{2, \overset{\checkmark}{4}, 3, 1, 5\}$

2nd Iteration $\{2, 3, \overset{\checkmark}{4}, 1, 5\}$

3rd Iteration $\{\boxed{2, 3, 1}, \boxed{4, 5}\}$

$\downarrow$ unsorted $\quad \downarrow$ sorted

✓ 3rd Ele at the End.

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

1st iteration $\{2, \overset{\checkmark}{3}, 1, 4, 5\}$

2nd iteration $\{\boxed{2, 1}, \boxed{3, 4, 5}\}$

$\downarrow$ unsorted $\quad \downarrow$ sorted

✓ 4th Ele at the End.

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

1st Iteration $\{\boxed{1}, \boxed{2, 3, 4, 5}\}$

$\downarrow$ unsorted $\quad \downarrow$ sorted

$\downarrow$

Sorted !

$j = 0 \longrightarrow N - i - 1$

$i = 1$
$j = \overset{}{0}, 1, 2, 3$ ✓ ✓

$i = 2$
$j = 0, 1, 2$ ✓ ✓ ✓

$i = 3$
$j = 0, 1$ ✓ ✓

$i = 4$
$j = 0$ ✓

Selection Sort.

int [] arr = { 5, 4, 1, 3, 2 } $\xrightarrow{\text{selection sort}}$ { 1, 2, 3, 4, 5 }

* Select smallest element from the unsorted array and place it at first pos of the unsorted part }.

step 1

$$\begin{array}{cccccc} & 0 & 1 & 2 & 3 & 4 \\ \text{arr} = \{ & 5, & 4, & \boxed{1}, & 3, & 2 \} \end{array}$$

0-4 → (i-1, N-1)

↳ unsorted part

step 2

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

1-4

sorted part

↳ unsorted part

step 3

$$\begin{array}{cccccc} & 0 & 1 & 2 & & \\ \{ & 1, & 2, & 5, & \boxed{3} & 4 \} \end{array}$$

2-4

sorted

↳ unsorted

step 4

$$\begin{array}{cccccc} & 0 & 1 & 2 & 3 & \\ \{ & 1, & 2, & 3, & 5, & \boxed{4} \} \end{array}$$

3-4

Sorted

↳ unsorted

{ 1, 2, 3, 4, 5 } ✓ Ans!

```
✓ for (int i=1 ; i<=N-1; i++)
  {
     int index=-1, minEle = Integer.MAX_VALUE;
     for(int j = i-1; j <=N-1; j++)            get minEle, and index
     {
          if(arr[j] < minEle)
          { minEle = arr[j];
            index = j;
          }
     }
     ⟹
     swap (arr[index], arr[i-1]);
  }
```

Q print value and index of smallest ele in a array?

$$\begin{array}{cccccccccc} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ arr = \{ & 3, & 10, & 4, & 2, & 9, & 1, & 20, & 25 \} \end{array}$$

O/p value : 1 , index = 5

minEle = ~~∞~~ ~~3~~ ~~2~~ 1
idx = ~~-1~~ ~~0~~ ~~3~~ 5

```
int minEle = Integer.MAX_VALUE;
int index = -1;
for(int i=0; i<n; i++)
{
    if(arr[i] < minEle)
    {
        minEle = arr[i];
        index = i;
    }
}
```

Sort A array

```
int[] arr = {1,2,5,7,9,11}
```

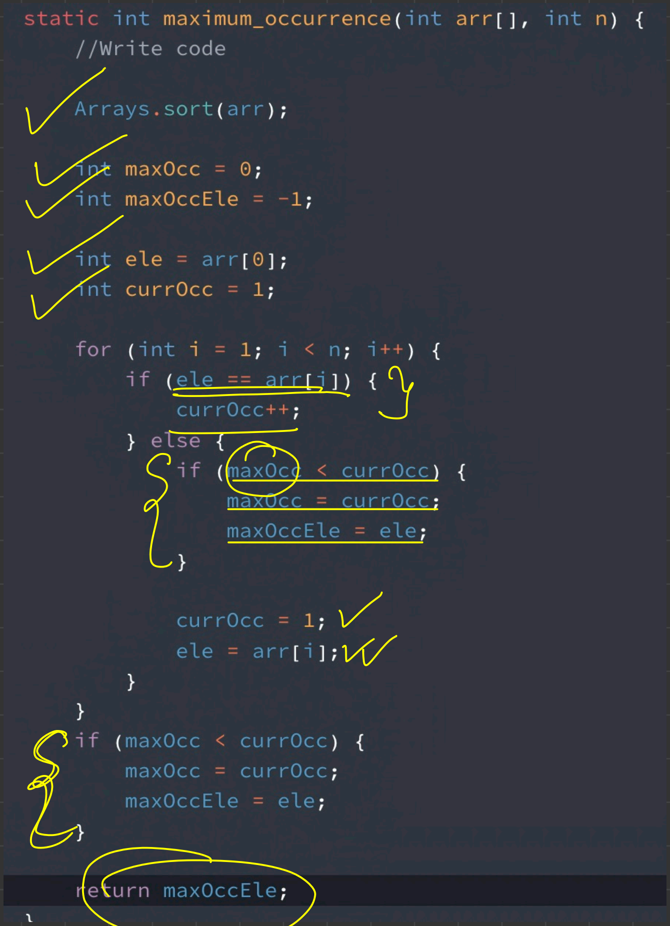
```
Arrays.sort(arr);
```

$\rightarrow$ TC: O(NlogN)
SC: O(1)

$\rightarrow$ Hybrid Sorting Algorithm

Arraylist

```
Collections.sort(list);
```



```java
static int maximum_occurrence(int arr[], int n) {
    //Write code

    Arrays.sort(arr);

    int maxOcc = 0;
    int maxOccEle = -1;

    int ele = arr[0];
    int currOcc = 1;

    for (int i = 1; i < n; i++) {
        if (ele == arr[i]) {
            currOcc++;
        } else {
            if (maxOcc < currOcc) {
                maxOcc = currOcc;
                maxOccEle = ele;
            }

            currOcc = 1;
            ele = arr[i];
        }
    }
    if (maxOcc < currOcc) {
        maxOcc = currOcc;
        maxOccEle = ele;
    }

    return maxOccEle;
}
```

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

= {1, 1, 1, 2, 3, 3, 3}

maxOcc = ~~0~~ 3

maxOccEle = ~~-1~~ 1

ele = ~~1~~ ~~2~~ 3

currOcc = ~~1~~ ~~2~~ ~~3~~ ~~1~~ ~~1~~ ~~2~~ 3