

Kunal M2

Topic

{ Time Complexity  
&  
Space Complexity

↳ Theory  
↳ Question

↳ what  
↳ time (logic)

↳ logic TC

→ ↳ Dry run

↳ Code

↳ Doubts

⇒ Sorting (arranging a sequence in a particular order)

- Bubble Sort
- Insertion Sort
- Selection Sort

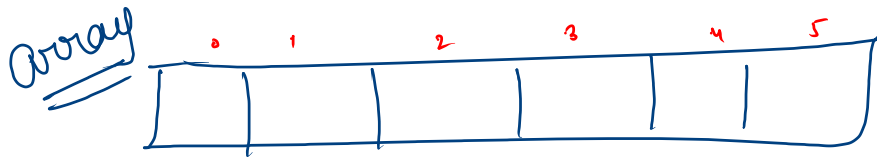
Time Complexity

$$O(N^2)$$

where,  $N$  is size of your sequence

Space Complexity

$$O(1)$$



$$N = 6, \quad \underline{\underline{T.C = 36}}$$

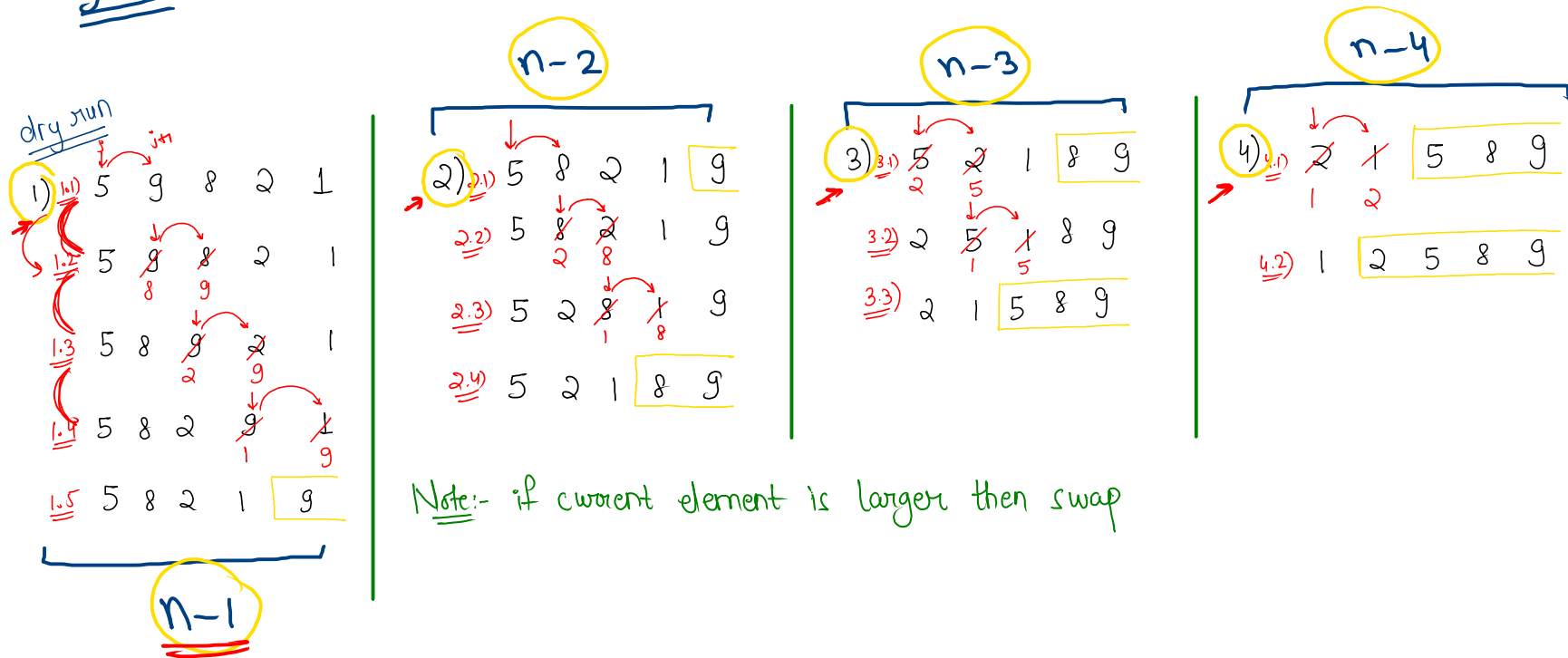
# ⇒ Bubble Sort

↳ pick the largest element and place it to the rightmost part of unsorted array

arr =  
size = n

5	9	8	2	1
0	1	2	3	4

→ collection of similar type of data



→ Nested loops

$i \leq n-1$   
 $i < n$

pseudo  
code

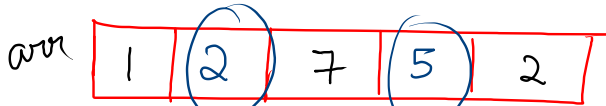
→ for ( i = 1 → n-1 ) {

for ( j = 0 → n-i ) {

if ( arr[j] > arr[j+1] )  
    swap( j, j+1 )

}

y



x=1  
y=3

temp = 5 ,

```
public static void main(String[] args) {  
    Scanner scn = new Scanner(System.in);  
    int n = scn.nextInt();  
    int[] arr = new int[n];  
    for (int i = 0; i < n; i++) {  
        arr[i] = scn.nextInt();  
    }  
    bubbleSort(arr, n);  
}
```

```
public static void bubbleSort(int[] arr, int n) {  
    for (int i = 1; i <= n - 1; i++) {  
        for (int j = 0; j < n - i; j++) {  
            if (arr[j] > arr[j + 1]) {  
                swap(arr, j, j + 1);  
            }  
        }  
    }  
}
```

```
// print  
for (int i = 0; i < n; i++) {  
    System.out.print(arr[i] + " ");  
}
```

```
public static void swap(int[] arr, int x, int y) {  
    int temp = arr[x];  
    → arr[x] = arr[y];  
    → arr[y] = temp;  
}
```

⇒ Insertion Sort (Pick the first element of unsorted array and place it in the correct position of sorted array)

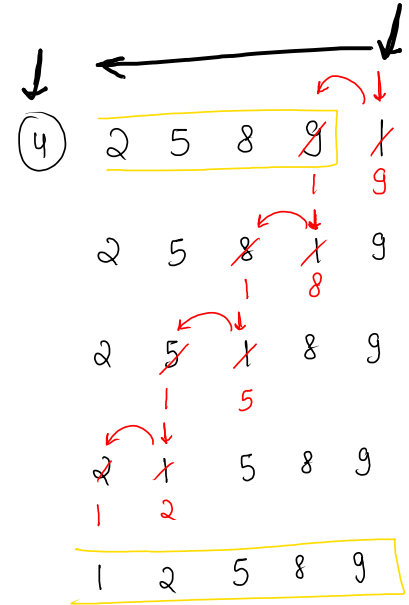
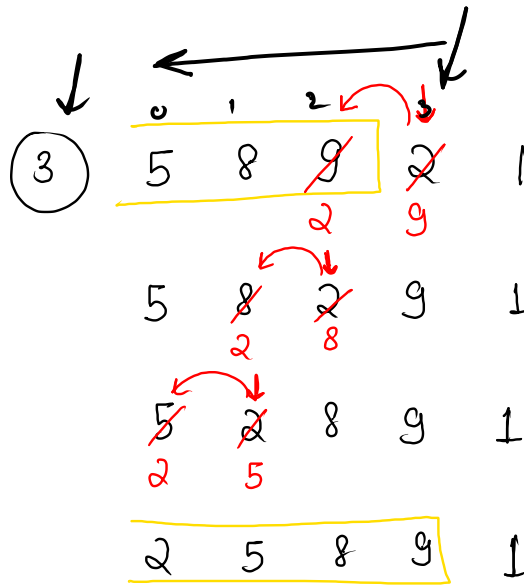
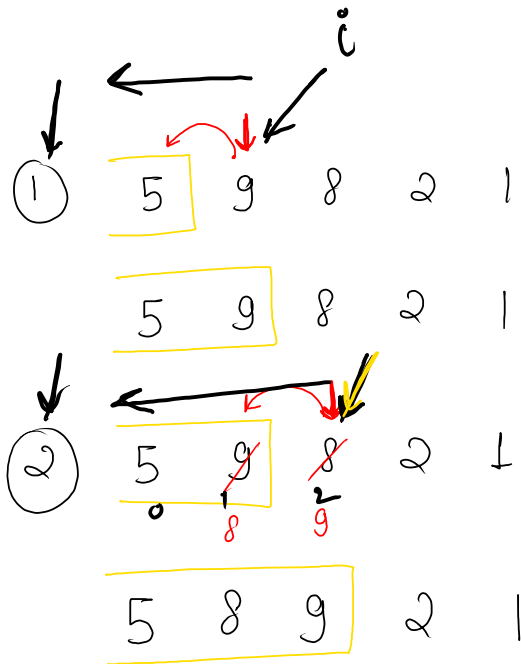
arr =

5	9	8	2	1
0	1	2	3	4

size = n

dry run

Note:- here, assume 1st element as already sorted.



Note:- if prev. element is larger then swap  
 $(j), (j+1)$

```

public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
    insertionSort(arr, n);
}

public static void insertionSort(int[] arr, int n) {
    for (int i = 1; i < n; i++) {
        for (int j = i - 1; j >= 0; j--) {
            if (arr[j] > arr[j + 1]) {
                swap(arr, j, j + 1);
            } else {
                break;
            }
        }
    }

    // print
    for (int i = 0; i < n; i++) {
        System.out.print(arr[i] + " ");
    }
}

public static void swap(int[] arr, int x, int y) {
    int temp = arr[x];
    arr[x] = arr[y];
    arr[y] = temp;
}

```

arr

0	1	2	3	4
2	3	4	5	7

$i=1, j=0$   $(5 > 2) \checkmark$

$j = -1$

$i=2, j=1$   $(5 > 3) \checkmark$

$j = 0$

$(2 > 3) \times$

$i=3, j=2$

$(5 > 7) \times$

$i=4, j=3$

$(7 > 4) \checkmark$

$j=2$

$(5 > 4) \checkmark$

$j=1$

$(3 > 4) \times$

$i=5 \times$

