

## Sorting

Arranging data in a specific order applied on one or more properties (attributes) of data

Name	Matches	Runs
Gopal	32	4100
Ankesh	25	1020
Vinod	20	1289
Sahil	16	2100

SPL

## Inbuilt functions

Sort()

TC:  $O(N \log N)$

SC: Depends on algo

Q

### Element Removal

Zomato

PayTM

Given an array of  $N$  elements. Remove all elements one-by-one.

Cost to remove an element = Sum of all the elements in the array before the removal.

Return the minimum possible cost to remove all elements.

A : <sup>0</sup>2, <sup>1</sup>1, <sup>2</sup>4  $\longrightarrow$  11

Remove 2  $\longrightarrow$   $2 + 1 + 4 \Rightarrow 7$

Remove 1  $\longrightarrow$   $1 + 4 \Rightarrow 5$

Remove 4  $\longrightarrow$   $4 \Rightarrow 4$

Total cost  $\Rightarrow 16$

Remove 4  $\longrightarrow$   $2 + 1 + 4 \Rightarrow 7$

Remove 2  $\longrightarrow$   $2 + 1 \Rightarrow 3$

Remove 1  $\longrightarrow$   $1 \Rightarrow 1$

Total cost  $\Rightarrow 11$

$A_0 \quad A_1 \quad A_2 \quad A_3$

Remove  $A_0 \Rightarrow A_0 + A_1 + A_2 + A_3$

Remove  $A_1 \Rightarrow A_1 + A_2 + A_3$

Remove  $A_2 \Rightarrow A_2 + A_3$

Remove  $A_3 \Rightarrow A_3$

---

Total cost  $\Rightarrow A_0 + 2 \cdot A_1 + 3 \cdot A_2 + 4 \cdot A_3$

---

- Step I • Sort the array in desc order  $N \log N$   
 Step II • Iterate over array & add contribution  $N$  of every term.

TC :  $O(N \log N)$

SC : Depends on the algo

Q

Nobel integer

Amazon

Direct-1

Given an array of size  $N$  having distinct elements.  
 Count the no. of nobel integers.

Nobel integer :

Count of elements smaller than it are equal to the element itself.

A : 

0 -1 2	1 -5 1	2 3 3	3 5 5	4 -10 0	5 4 4
--------------	--------------	-------------	-------------	---------------	-------------

 $\rightarrow 3$

# smaller elements

$-10 \quad -5 \quad -1 \quad 3 \quad 4 \quad 5$

## Brute Force

TC:  $O(N^2)$

SC:  $O(1)$

For every element:

Iterate over the array &  
Count the no. of smaller  
elements.

if (count == element)  
ans++

Step I • Sort the array in asc order  $O(N \log N)$

Step II • Iterate over the array  $O(N)$   
if ( $A[i] == i$ )  
ans++

TC:  $O(N \log N)$

SC: Depends on the algo

HW What if : Repetition is allowed?

A : 

	<sup>0</sup>	<sup>1</sup>	<sup>2</sup>	<sup>3</sup>	<sup>4</sup>	<sup>5</sup>	<sup>6</sup>
	3	4	4	5	5	5	10
	0	1	1	3	3	3	6

A : 

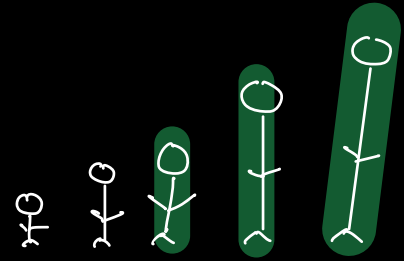
	<sup>0</sup>	<sup>1</sup>	<sup>2</sup>	<sup>3</sup>	<sup>4</sup>	<sup>5</sup>
	0	2	2	3	3	6
	0	1	1	3	3	5

## Bubble Sort

A : 4, 8, 6, 2, 3

At max  
N Times

4	6	2	3	8
4	2	3	6	8
2	3	4	6	8
2	3	4	6	8



ASC

```
for (i=0; i<N; i++){
```

```
    for (j=0; j<N-1 ; j++){
```

```
        if (A[j] > A[j+1]){
            swap (A[j], A[j+1]);
        }
```

```
    }
```

```
}
```

i  
0  
1  
2  
3

j  
j < N-1  
j < N-2  
j < N-3  
j < N-4

}

$j < N-1-i$

TC :  $O(N^2)$

SC :  $O(1)$

DESC

```
for (i=0; i<N; i++) {
```

```
    for (j=0; j<N-1; j++) {
```

```
        if (A[j] < A[j+1]) {
            swap (A[j], A[j+1]);
        }
```

```
    }
```

```
}
```

```
for (i=0; i<N; i++) {
```

```
    for (j=0; j<N-1; j++) {
```

```
        if (Comp(A[j], A[j+1])) {
            swap (A[j], A[j+1]);
        }
```

```
    }
```

```
}
```

9, 2, 6, 1, 12

Sort in ASC order based on count of factors.

A[i]	# factors
------	-----------

9	3
---	---

2	2
---	---

6	4
---	---

1	1
---	---

12	6
----	---

1 2 9 6 12

```
boolean comp ( int a, int b ) {
    // returns true if factorCount(b) <
    factorCount(a)
```

```
    int fa = factorCount(a);
    int fb = factorCount(b);
```

```
    if ( fb < fa )
        return true;
    else
        return false;
```

```
}
```

C++/JS

```
int CustomComparator ( int a, int b ) {
```

```
    if you want a before b → -1
    if you want b before a → 1
    if you want b == a → 0
```

```
}
```

```
Sort (A, CustomComparator)
```

Python

```
def compare (a, b):
```

```
    if you want a before b → -1
    if you want b before a → 1
    if you want b == a → 0
```

```
A.sort (key = cmp_to_key(compare))
```

C++ / C#

```
bool comp (a, b) {
```

if you want a before b  $\rightarrow$  true

else false

```
}
```

```
Sort (A, comp);
```