Sorting ⇒ Arranging elements in order based on a property.

Eg: 1) Words in a Dictionary
2) Books in a library.

Integers.

[-5, 10, 2, 3, 0]

[-5, 0, 2, 3, 10] → Increasing
[10, 3, 2, 0, -5] → Decreasing.

[-5, 10, 0, -5, 0]

[-5, -5, 0, 0, 10] → Non decreasing
[10, 0, 0, -5, -5] → Non increasing

Ascending
Descending.

Characters

→ 1 Byte → 8 bits → $2^8 = 256$
$\downarrow$
[0, 255]

a = 97	A = 65
b = 98	B = 66
⋮	⋮
z = 122	Z = 90

ASCII values

'a' < 'c'

'A' < 'a'

Strings   Alphabetic order / Lexographical

1) a l l
2) a p e
3) a p p l e

Custom Comparisom

1) Arrange the given strings in ascending order of length.

[apple, bat, dog, carrot, it]

5 3 3 6 2

[it, bat, dog, apple, carrot]

2) Arrange integers in descending order of their absolute value.

$|-2| = 2$

[-10, 6, 2, -5, 0, 8]

10 6 2 5 0 8

$x \to x$ if +ve

$ \to -x$ if -ve

[-10, 8, 6, -5, 2, 0]

Sorting functions

→ Array List

JAVA ⇒ Collections. sort(al)

Arrays. sort(arr)

→ array.

C++ ⇒ sort(a. begin(), a. end())

Python ⇒ A. sort()

A = sorted(A)

JS ⇒ A. sort()

T.C. = $O(N \log N)$

S.C. = $O(1)$

(for intermediate only)

Stable Sort

⇒ Relative order of equal elements should not change.

Q

Given an integer array of size N. Find the min cost of removing all elements from the array.

Cost of removing an element = Sum of all elements before removing this element.

$A = [2, 4, 1]$

Case 1

Ele	Cost	
2	2+4+1 = 7	[4, 1]
4	4+1 = 5	[1]
1	1 = 1	[ ]
	13	

Case 2

Ele	Cost	
4	2+4+1 = 7	[2, 1]
1	2+1 = 3	[2]
2	2 = 2	
	12	

Case 3	Ele	Cost	
	4	2 + 4 + 1 = 7	[2, 1]
	2	2 + 1 = 3	[1]
	1	1 = 1	[]
		11	

Observations

↳ Remove elements in decesending order.

$[a, b, c, d]$

Ele	Cost	
a → Largest	$a + b + c + d$	$[b, c, d]$
b	$b + c + d$	$[c, d]$
c	$c + d$	$[d]$
d → Smallest	d	$[]$

Descending. $a + 2b + 3c + 4d$

Goal: Cost of removing elements in descending order.

$$A = \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ [10, & 8, & 2, & 0, & -1] \\ 1 & 2 & 3 & 4 & 5 \end{array}$$

A[i] ⇒ (i+1) times.

Code

```
ans = 0
for (i = 0; i < N; i++) {
        ans += A[i] × (i+1);
}
return ans;
```

$$T.C. = O(N\log N + N)$$
$$= O(N\log N)$$

Q. Given an integer array of size N. Count the no. of noble integers present. Distinct elements.

Noble integer ⇒ Count of elements $< A[i]$ = A[i]

$$A = \begin{array}{ccccc} \checkmark & \checkmark & \checkmark & \checkmark & \checkmark \\ 0 & 1 & 2 & 3 & 4 \\ \left[2, \right. & 4, & 0, & 1, & \left. 3\right] \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 2 & 4 & 0 & 1 & 3 \end{array} \qquad Ans = 5$$

$$A = \begin{array}{ccccc} \checkmark & \checkmark & \checkmark & \checkmark & \checkmark \\ 0 & 1 & 2 & 3 & 4 \\ \left[0, \right. & 0, & 0, & 0, & \left. 0\right] \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 0 & 0 & 0 & 0 & 0 \end{array} \qquad Ans = 5.$$

<u>Sol^n</u>

1) <u>Brute Force</u>

⇒ $\underset{O(N)}{\underline{\forall i}}$, iterate & count no. of elements < A[i]

$$T.C. = O(N^2)$$

$$A = \begin{array}{ccccc} \left[-2, \right. & 0, & \boxed{2}, & 5, & \left. \boxed{3}\right] \\ 0 & 1 & \boxed{2} & 4 & \boxed{2} \end{array}$$

2) <u>Optimise</u>

Count of no.s less than any element.

⇒ Sorted.

$$A = \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ -2, & 0, & \bigcirc{2}, & 3, & 5 \end{array}$$

$$\underbrace{0 \quad (i-1)}_{<} \quad \underline{i}$$

$\# = i$

Code

```
count = 0;
A.sort();
for (i=0; i<N; i++) {
    if (i == A[i]) {
        count++;
    }
}
return count;
```

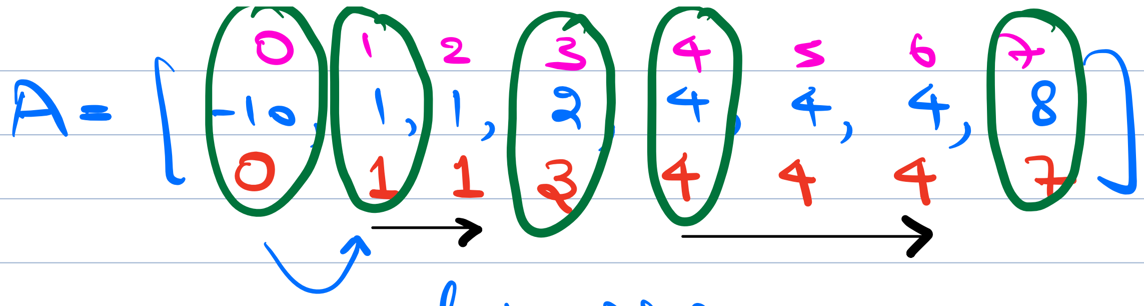
T.C. = $O(N \log N)$
S.C. = $O(1)$

Q If array has duplicate elements.

$$A = \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \\ -10, & 1, & 3, & 1, & 100 \end{array}$$

$$A = \begin{array}{ccccc} -10, & 1, & 1, & 3, & 100 \\ 0 & 1 & 1 & 3 & 4 \end{array}$$

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



prev ele != A[i]

Code

```
A.sort();
ans = 0;
count = 0;          if (A[0] == 0) { ans++; }

for (i = 1; i < N; i++) {
        if (A[i] != A[i-1]) {
                count = i;
        }
        if (count == A[i]) {
                ans++;
        }
}
return ans;
```

T.C. = $O(N \log N)$

S.C. = $O(1)$

Q Given an integer array of size N (all +ve)

Sort the array (ASC) on the basis of count of factors.

$A[i] <<< N$

$$A = \left[\begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 7, & 13, & 9, & 12, & 36, & 16, & 1 \\ 2 & 2 & 3 & 6, & 9, & 5 & 1 \end{array}\right]$$

$$A = [1, 7, 13, 9, 16, 12, 36]$$

Custom Comparator

int compare (x, y) {

return → -ve (x should be on left of y)

→ 0 (x = y ⇒ Keep x, y in same order)

→ +ve (y should be on left of x)

```
}
```

CC for factor Question.

```
A. sort (custom);

int custom (x, y) {   //O(1)
    int countx = factor(x);
    int county = factor(y);     > √A[i]
    if (countx < county) {
        // x to come on left of y.
        return -1;
    }
    else if (countx == county) {
        return 0;
    }
    else {
        return 1;
    }
}
```

T.C. = $O(N \log N)$