

Welcome 😊

Agenda : Sorting why, When
2-3 problems.
Comparator funcⁿs.

Sorting → arrangement of data in particular order on the basis of some parameter.

eg, 2 3 9 12 17 19 → sorted in asc. order of magnitude

eg: 19 6 5 2 -1 -19 → sorted in desc. order of magnitude.

Quiz

	1	13	9	6	12	Sorted in asc. order of count of factors.
count of factors	1	2	3	4	6	

Why sorting?

Searching becomes easier.

Finding data becomes easier.

How sorting ~~advance~~

Prbuit Sort function $\Rightarrow n \log n$
Sort()

When sorting

$O(n)$
↓
no need of sorting.

$O(n^2)$
↓ sorting
 $O(n \log n)$

\Rightarrow Always think of sorting, in case you are stuck in a question.

Q1 Given an array of integers N , you have to delete all the elements from the array.
You have to pay some cost to delete an element.

cost = sum of all elements present in the array. at the point

Find the min cost.

eg: $A: [2, 1, 4]$

$$\text{delete } 1 \rightarrow 2 + 1 + 4 = 7$$

$$\text{delete } 2 \rightarrow 2 + 4 = 6$$

$$\text{delete } 4 \rightarrow 4 = 4$$

$$\underline{\underline{17}}$$

min. cost

$$\text{delete } 4 \rightarrow 2 + 1 + 4 = 7$$

$$\text{delete } 2 \rightarrow 2 + 1 = 3$$

$$\text{delete } 1 \rightarrow 1 = \underline{\underline{1}}$$

Q2 eg: $[4, 6, 1]$

$$\text{delete } 6 \rightarrow 6 + 4 + 1 = 11$$

$$\text{delete } 4 \rightarrow 4 + 1 = 5$$

$$\text{delete } 1 \rightarrow 1 = \underline{\underline{17}}$$

Ques

$$[\overset{\textcircled{2}}{3} \overset{\textcircled{1}}{5} \overset{\textcircled{3}}{1} -3]$$

$$\text{delete } 5 \Rightarrow 3 + 1 + (-3) = 1$$

$$\text{delete } 3 \Rightarrow 3 + (-3) = 0$$

$$\text{delete } 1 \Rightarrow (-3) = -3$$

$$\text{delete } -3 \Rightarrow -3 = \underline{\underline{-3}}$$

$$[a \quad b \quad c \quad d]$$

$$\text{remove } a = a + b + c + d$$

$$\text{remove } b = b + c + d$$

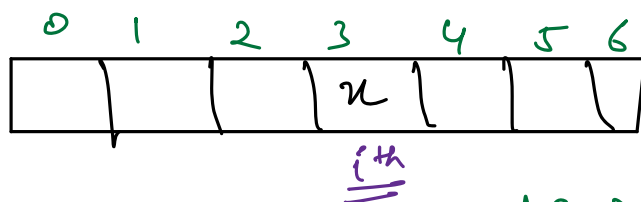
$$\text{remove } c = c + d$$

$$\text{remove } d = d$$

$$\underline{a + 2b + 3c + 4d}$$

$$a > b > c > d$$

Ans: Sorting the data can help in getting largest element each and every time.



dec order

$(i+1)$ contribution while deleting $i\text{-th}$ element.

Pseudo code

1) sort data in desc. order $\xrightarrow{\text{s.c. } O(N)}$ $n \log n$

2) $ans = 0$
for ($i=0$; $i < n$; $i++$) } $\rightarrow n$
{
 $ans += (i+1) * arr[i]$;
}

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

S.C $O(N)$
merge sort } $O(N)$
quick sort } space.

Q2 Find count of noble integers.

array size N

↓
distinct elements.

$A[i]$ is noble if

count of elements smaller than $A[i]$ = $A[i]$

eg :

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

ans = 3

eg :

-3	0	2	5
↓	↓	↓	↓
0	1	2	3

ans = 1

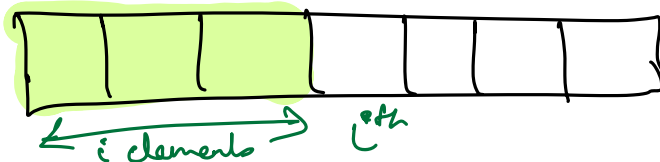
Brute force

① for every element, count smaller elements.

T.C $\Rightarrow O(N^2)$

Optimized

sort data (asc)



Smaller i^{th} element $\rightarrow b - a + 1$
 $(i-1) - 0 + 1$
 $= i$

Pseudo

```
① sort the data
② for ( i = 0 ; i < n ; i++)
{
    if ( arr[i] == i )  → count of smaller elements.
        ans++;
}
return ans;
```

What if elements are duplicate?

eg:

-10	1	1	3	100	<u>ans = 3</u>
↓	↓	↓	↓	↓	
0	1	<u>1</u>	<u>3</u>	4	

count of smaller element

eg:

-10	1	1	2	4	4	4	8	10
↓	↓	↓	↓	↓	↓	↓	↓	↓
0	1	1	3	4	4	4	7	8

count of smaller element

ans = 5

Pseudo

```
if ( curr element != prev elem )
    count = index.
```

```
① Sort the data . cnt = 0
② if ( arr[0] == cnt ) ans++;
for ( i = 1 ; i < N ; i++)
{
    if ( arr[i] != arr[i-1] )
        cnt = i;
    if ( cnt == arr[i] )
        ans++;
}
return ans;
```

T.C → $n \log n$

Q Sort data in asc order of count of factors.

If count of factors are equal, then sort on basis of magnitude.

eg:

9	3	10	6	4
↓	↓	↓	↓	↓
3	2	4	4	3

sorted data

3	4	9	6	10
---	---	---	---	----

sort() → by default asc order of magnitude
X

sort (— — —)
↓ ↓ ↓
start end Index

→ comparator funcⁿ
i.e. rules of sorting.

x, y

$\text{countFactor}(x) < \text{countFactor}(y)$

x should come first

$\text{countFactor}(x) > \text{countFactor}(y)$

y should come first

$\text{countFactor}(x) == \text{countFactor}(y)$

if ($x \leq y$)

x should come first

if ($x > y$)

y should come first

★★

if first argument
should be returned
in sorted data

return True

else
return False.

datatype of
element that you
are sorting.

```
bool comp (int n, int y)
{
    int cntX = countFactors(n);
    int cntY = countFactors(y);

    if (cntX < cntY)
        return true;
    if (cntX > cntY)
        return false;

    else {
        if (n <= y)
            return true;
        else
            return false;
    }
}
```

Q Sort data in desc. order of magnitude.

```
bool comp (int n, int y)
{
    if (n >= y)
        return true;
    else
        return false;
}
```


Q

array of strings

sort acc. to length of string. in asc. order.

```
bool comp (string u, string v)
```

```
{
```

```
    if ( u.length() <= v.length() )
```

```
        return true.
```

```
    else
```

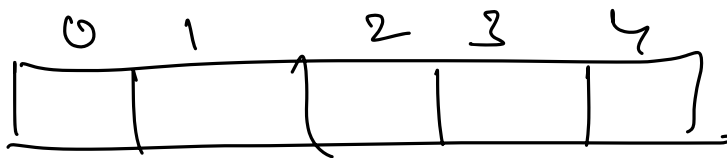
```
        return false
```

```
}
```

```
sort ( start Index, end Index, comp() )
```

Doubt session

```
sort ( 0, 4, comp() )
```



Q

$$(a \% p * b) \% p = (a * b) \% p$$

$$t = (10^0) \% p$$

$$t = (t * 10) \% p \Rightarrow$$
$$= t \% p$$

$$\frac{(10^2)}{t} = (t \times 10) \div 29$$