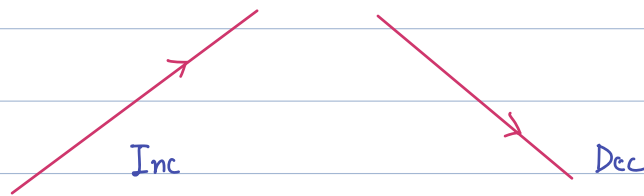


Friday : Contest 2

{ Bit Manipulation, sorting, modular arithmetic, string }

Sorting : Arranging data in increasing or decreasing order based on a parameter

Cricketers	Salary	Runs
MSD	1Cr ^③	7000
Virat	5Cr ^④	10000
SKY	0.5Cr ^①	3000
Jadeja	1Cr ^②	2500



{ 3 5 7 9 10 } Inc

based on value

{ 1, 13, 9, 6, 12 } Inc

factors: 1 2 3 4 6 based on factors

Algorithms

- | | | |
|----|----|----------------|
| 1) | A1 | Merge sort |
| 2) | A2 | Quick sort |
| | ⋮ | Selection sort |
| | | ⋮ |

Advanced

Problem solving based on sorting

How to sort an array in your language?

	Python	C++	Java
$l =$	$l.sort()$	$sort(l, l+n)$	$Collection.sort(l)$
	↓ Inc		

TC: $O(N \log N)$
SC: $O(1)$

Q1) Given an array of N elements. Remove all elements from array one by one.

Cost of removing any element:

Sum of all array elements at that time (before deletion)

Return minimum cost of deleting all elements.

	0	1	2		0	1	2
A:	2	1	4	A:	2	1	4
Deleted 1:	$2+1+4 = 7$			4:			
Delete 2:	$2+4 = 6$						
Delete 4:	$4 = 4$						

Total cost: 17

$\{4, 6, 1\}$

Del 6: $6+4+1 = 11$

Del 4: $4+1 = 5$

Del 1: $1 = 1$

TC: 17

Remove the max element.

↓
2nd largest \rightarrow 3rd largest \rightarrow Smallest

Why?

70

$$\{ \overset{\times}{a_0} \quad a_1^{\times} \quad a_2 \quad \overset{\times}{a_3} \} \quad 4$$

$$\text{Del } a_0: \quad a_0 + a_1 + a_2 + a_3 \quad a_0$$

$$\text{Del } a_1: \quad a_1 + a_2 + a_3 \quad a_1$$

$$\text{Del } a_3: \quad a_2 + a_3 \quad a_2$$

$$\text{Del } a_2: \quad a_2 \quad a_3$$

$$a_0 + 2a_1 + 4a_2 + 3a_3$$

Minimize

$$a_3 < a_2 < a_1 < a_0$$

$$4a_3 + 3a_2 + 2a_1 + a_0$$

Higest coefficient needs to be multiplied by
least value

$$a_2 < a_3 < a_1 < a_0$$

Obs: Del elements in decreasing order

1st lowest element $\times N$

2nd lowest $\times N-1$

\vdots

largest $\times 1$

l.sort() // asc order

mincost = 0

TC: $O(N \log N + N)$

temp = N

SC: $O(1)$

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

 mincost += temp * A[i]

 temp --

}

return mincost

countlgs:

0	1	2	3
0	1	2	3
0	1	2	3

Amazon

Q2) Nobel integer

Given an array of size N having distinct elements. Count number of nobel integers.

unique elements

Nobel integer: Any element $A[i]$ for which
count of elements less than it = $A[i]$

A:	0	1	2	3	4	5
	-1	-5	3	5	-10	4
Countless	2	1	3	5	0	4

Nobel : 3

A:	0	1	2	3	4	5	6	7
	4	8	3	2	-1	1	7	6

$\{ -3, 0, 2, 5 \}$
 countless 0 1 2 3

$\boxed{\text{nobel} = 1}$

Brute force: For every index i

TC: $O(N^2)$

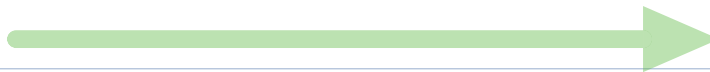
count no. of $\text{eles} < A[i]$

count == $A[i]$

nobel

Optimised approach:

0 1 2 3 4 5 6 7
 4 8 3 2 -1 1 7 6
 sort
 0 1 2 3 4 5 6 7
 -1 1 2 3 4 6 7 8
 countless: 0 1 2 3 4 5 6 7



$\boxed{\text{countless} = i}$

l.sort() nobel = 0

TC: $O(N \log N)$

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

if ($A[i] == i$) {

nobel++

}

3

Data can Repeat $\{ \text{Nobel integers} \}$

0 1 2 3 4 5
Ex: $\{ 0, 2, 2, 3, 3, 6 \}$

0 1 2 3 4 5 6 7 8
Ex: $\{ -10, 1, 1, 1, 4, 4, 4, 7, 10 \}$ -

0	1	2	3	4	5	6	7	8
Ex: $\{ -10 \}$	1	1	2	4	4	4	8	10
counts:	1	1	3	4	4	4	7	8

Ans: 5

	0	1	2	3	4	
{	-10	1	1	3	100	}
countless:	0	1	1	3	4	

output: 3

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	
{	-3	0	2	2	5	5	5	5	8	8	10	10	10	14	}
countless:	0	1	2	2	4	4	4	4	8	8	10	10	10	13	

Ans: 7

1) Brute force remains same

2) countless = i \propto

✓ obs 1) countless remains same for equal values

✓ 2) First occurrence countless = i

AT[i] is repeating or first occurrence?
 \rightarrow check previous value

l.sort()

nobel = 0, countless = 0

```
for (i = 0; i < N; i++) {  
    if (A[i] != A[i-1]) {  
        countless = i  
        if (countless == A[i]) {  
            nobel++  
        }  
    }  
}
```

TODO: fix this
code

10:39 - 10:50

Comparators

[x, y, z]

Compare x and y

$x < y$

Compare y and z

$y < z$

[x, y, z]

Q Given an array, sort the array based on number of factors? if no. of factors are equal sort basis their value

0 1 2 3 4 5 6
 $\{ 9, 3, 4, 8, 16, 37, 6 \}$

factors: 3 2 3 4 5 2 4

Sorting: $\{ 3, 37, 4, 9, 6, 8, 16 \}$

int compare(a, b) {

cf-a = countfactors(a)

cf-b = countfactors(b)

if (cf-a < cf-b) {

return -1

if (cf-a > cf-b) {

return 1

if (cf-a == cf-b) {

if (a < b) {

return -1

if (a > b) {

return 1

if (a == b) {

return 0

}

sort() // sorts in asc order

sort() ← attribute??
comparators

comparators: It is a function with help of which we can override and sort data basis a given attribute

Rule:

- 1) Comparator takes 2 parameters as input
- 2) returns 0, -1, 1
- 3) If you return -1 (first parameter will appear before second parameter)
- 4) If you return 1 (2nd parameter will appear before first parameter)
- 5) If you return 0 (anything can come first)

increasing order

{ 1 , 3 , 5 , 7 }

3

5

int compare (a , b) {

if (a < b) {

|

return -1

}

if (a > b) {

|

return 1

}

if (a == b) {

return 0

}

sort (compare)

language dependent

syntax

for decreasing order

```
int compare(a, b) {  
    if (a < b) {  
        return 1  
    }  
    if (a > b) {  
        return -1  
    }  
    if (a == b) {  
        return 0  
    }  
}
```

TC: $O(N \log N) * \text{TC}(\text{comparator method})$

ChatGPT

how to write comparator

Rapid fire

Q) How to check if array is sorted in asc ord?

{ 5 3 7 9 11 }



$$\underline{A[i] \geq A[i-1]}$$

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

Q Array check if it is an AP?

return

{ 3 6 0 -3 -6 9 }

sort

{ -6 -3 0 3 6 9 }

sort()

$d = A[1] - A[0]$

for ($i=1; i < N; i++$) {

if ($A[i] - A[i-1] \neq d$) {
return false
}

$$\left[\begin{array}{|c|c|c|} \hline 4 & 2 & 13 \\ \hline \end{array} \right] \quad n^2$$

$$\{ 7 \quad 9 \quad 3 \quad 2 \}$$