Sorting
- why sorting
- Min cost to remove all elements
- Nobel Integer → version 1, → version 2
- comparator

Sorting → Arranging data in ↑ or ↓ order based on a parameter

Ex 1: 3 8 9 14 19 → value

Ex 2: 19 14 9 8 3 → value

Ex 3: 1 5 3 9 6 10 12 → no. of factors

1	5	3	9	6	10	12
↓	↓	↓	↓	↓	↓	↓
1	2	2	3	4	4	6

Do you know a predefined sort function in your language?

→ sort( )

O(n log n) Time — why? — How? → Advanced class.

You can use inbuilt sort function

Q1. Given N array elements, at every step remove an array element.

Cost to remove an element – sum of array elements present before removing the element.

Find the min cost to remove all the elements.

Ex 1: { 2, 1, 4 }

Remove 2 ⇒ { 2, 1, 4 } → 7
Remove 1 ⇒ { 1, 4 } → + 5
Remove 4 ⇒ { 4 } → + 4
{ } = 16

Remove 4 ⇒ { 2, 1, 4 } → 7
Remove 2 ⇒ { 2, 1 } → + 3
Remove 1 ⇒ { 1 } → + 1
= 11

Ex 2: { 4, 6, 1 }

Remove 6 ⇒ { 4, 6, 1 } → 11
Remove 4 ⇒ { 4, 1 } → 5
Remove 1 ⇒ { 1 } → 1
= 17

Ex 3: { 3, 5, 1, -3 }

Remove 5 ⇒ { 3, 5, 1, -3 } ⇒ 6
Remove 3 ⇒ { 3, 1, -3 } ⇒ 1
1 ⇒ { 1, -3 } ⇒ -2
-3 ⇒ { -3 } ⇒ -3
= 2

Observation = Sort the array in decreasing order

Why?

Ex: { a b c d }

Remove a ⇒ { a b c d } ⇒ a + b + c + d
Remove b ⇒ { b c d } ⇒ b + c + d
Remove c ⇒ { c d } ⇒ c + d
Remove d ⇒ { d } ⇒ d

1a + 2b + 3c + 4d

Can I say we should starting removing from the max element

{ 4 , 1 , 6 , 2 }

Remove 6 ⇒ 4 + 1 + 6 + 2

4 ⇒ 4 + 1 + 2

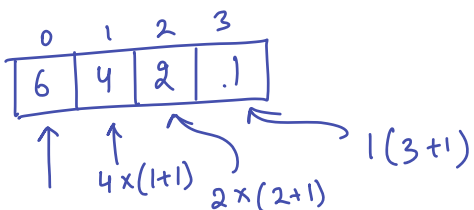
2 ⇒ 1 + 2

1 ⇒ 1

6(1) + 4(2) + 2(3) + 1(4)

Pseudocode

O(nlogn) → sort the array in decreasing order



O(n)

```
ans = 0
for(int i=0; i<n; i++){
    ans = ans + (arr[i] *(i+1))
}
return ans.
```

TC → O(nlogn) + O(n)

O(nlogn)

SC → O(1)

Q2. Given N elements, calculate no. of noble integers present.

Duplicates are not present

arr[i] is said to be noble integer if

{No. of elements < arr[i]} = arr[i]

Ex: { 1, -5, 3, 5, -10, 4 }

#count less → 2, 1, 3, 5, 0, 4

ans → 3

Quiz

{ -3, 0, 2, 5 }

#count less → 0, 1, 2, 3

ans = 1

Quiz

{ -10, -5, 1, 3, 4, 5, 10 }

#count less → 0, 1, 2, 3, 4, 5, 6

ans = 3

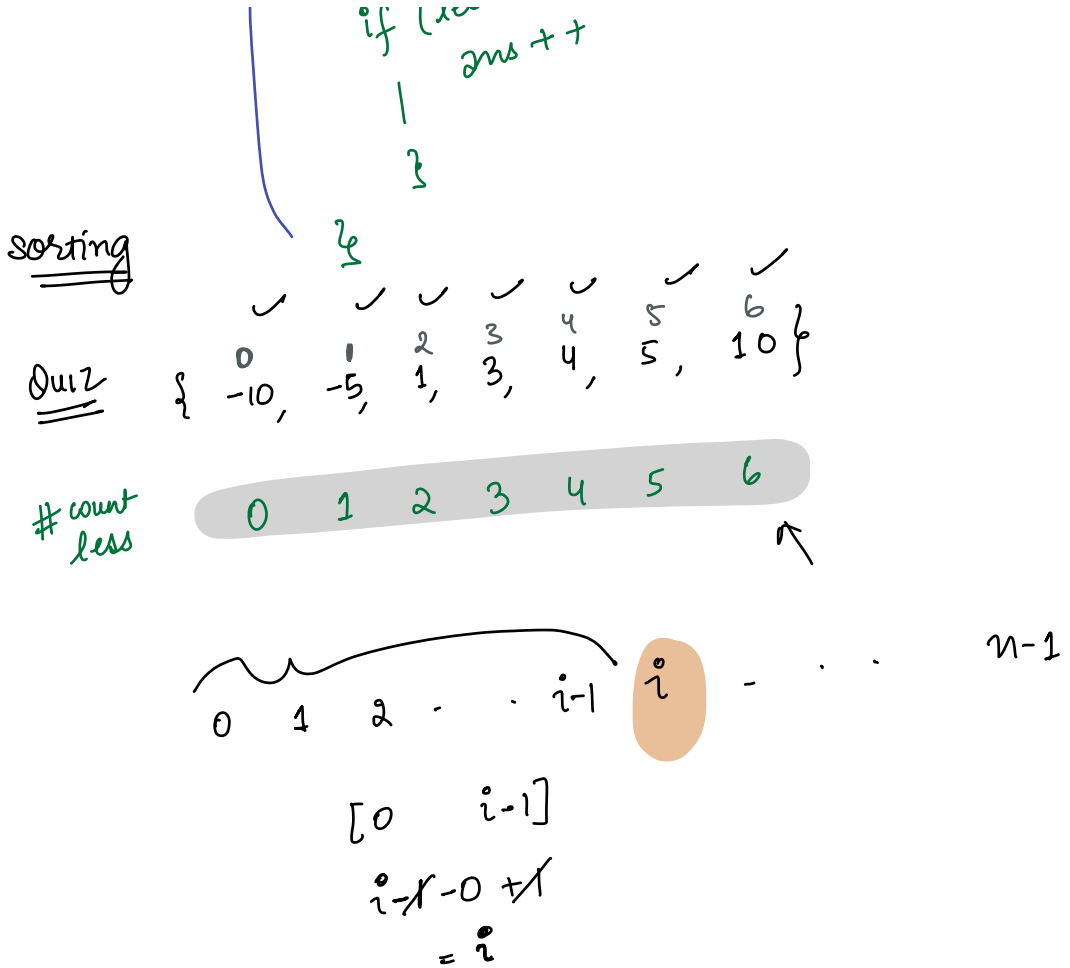
Simple Approach

```
ans = 0
for ( i=0 ; i<n ; i++){
    less = 0
    for( j=0 ; j<n; j++ ){
        if ( A[j] < A[i]){
            less++
        }
    }
    if ( less == A[i]){
```

TC → O(n²)

SC → O(1)

nos. less than A[i] is taking time



For an element,
number of elements lesser than A[i] is equal to i

Now we just need to check if A[i] == i

① Sort the array

② ans = 0
for (i=0; i<n; i++) {
  if (A[i] == i) {
    ans++
  }
}
return ans

TC → O(n log n) + O(n)
SC → O(1)

Version 2 : Elements can repeat

no of ele < arr[i]

→ Ex 1 : { 0 2 2 4 4 6 }

count less : 0 1 1 3 3 5

ans = 1

Ex 2 : { -10 1 1 3 100 }

count less : 0 1 1 3 4

ans = 3

Ex 3 : { -10 1 1 2 4 4 4 8 10 }

count less : 0 1 1 3 4 4 4 7 8

ans = 5

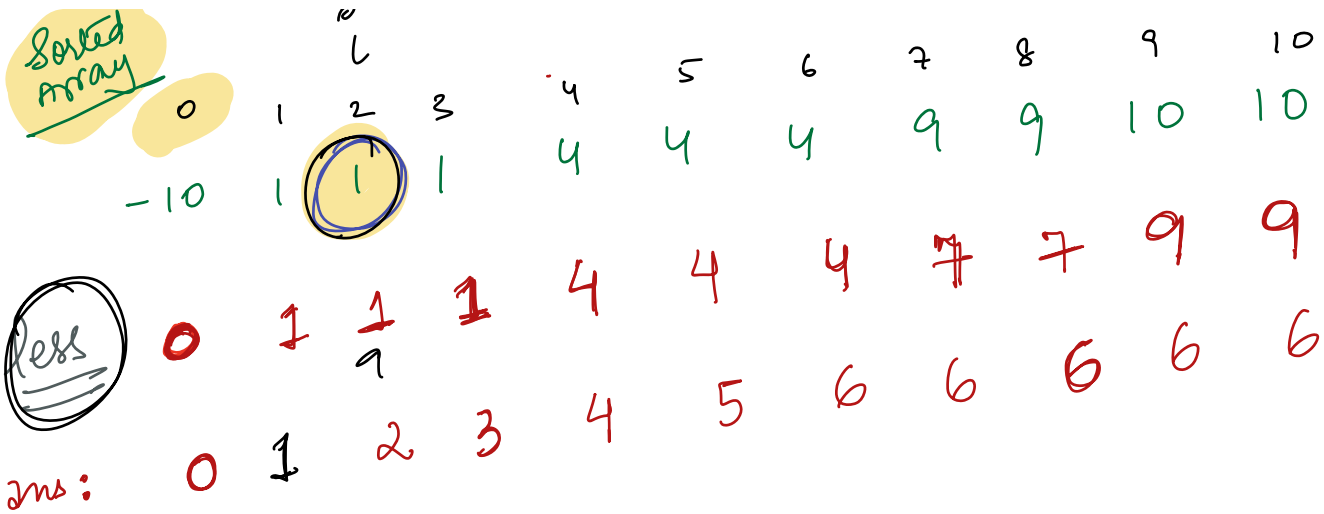
Ex : [-3] [0] [2 2] [5 5 5 5] [8 8] [10 10 10] [14]

count less : 0 1 2 2 4 4 4 4 8 8 10 10 10 13

ans = 7

For repeating elements, count is same

For first occurrence of each element, the count of elements less than A[i] = 2.



Q. How to know about the first occurrence?

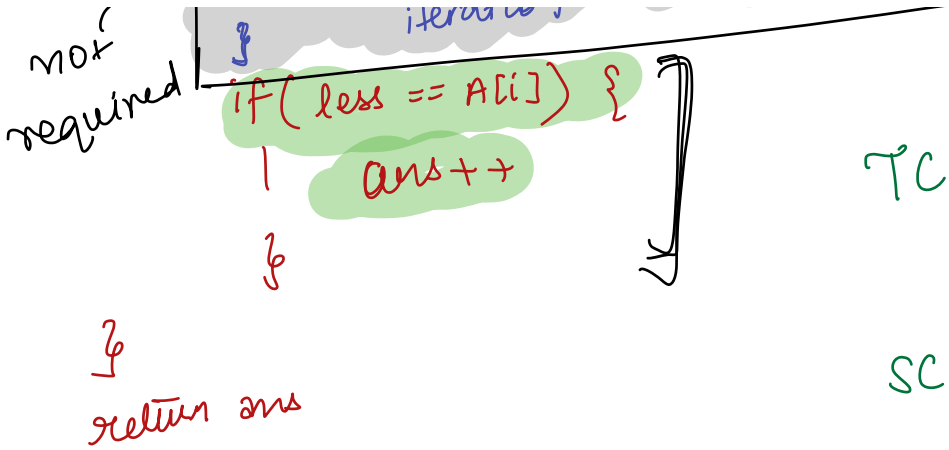
```
if ( A[i] != A[i-1] )
```

Pseudocode

① Sort the array

② less = 0 ans = 0

```
if ( A[0] == 0 ) ans++ ;
for ( i = 0/1 ; i < n ; i++ ) {
    if ( A[i] != A[i-1] ) { // first occurrence
        less = i
    }
    else { // not the first occurrence
        // less will be same as in previous
```



TC → O(n log n) + O(n)

SC → O(1)

Comparator

Given N array elements, sort them in increasing order of their no. of factors

If two ele have same factors, ele with less value will come first.

[No extra space is allowed]

Ex 1: {9, 3, 4, 8, 16, 37, 6, 13, 15}

0	1	2	3	4	5	6	7	8
9	3	4	8	16	37	6	13	15
↓	↓	↓	↓	↓	↓	↓	↓	↓
3	2	3	4	5	2	4	2	4

{3, 13, 37, 4, 9, 6, 8, 15, 16}

Do we know any sorting technique — NO!

INBUILT sorting function

Modify the inbuilt sorting function

Ex: 25 16 → 25 should be placed first
3 5

Ex: 10 9 → 9 should be placed first
4 3

Ex: 49 9 → Factors same → compare values 9 < 49 → Put 9 first.
3 3

mergesort C++

```
bool comp(int a, int b){
// Note: If we want a to come before b in final array, then function should return true
// Q: When do u want a to come before b?
int f1 = factors(a)
int f2 = factors(b)
if (f1 < f2) return true
else if (f1 == f2 && a < b) return true
else return false
}
```

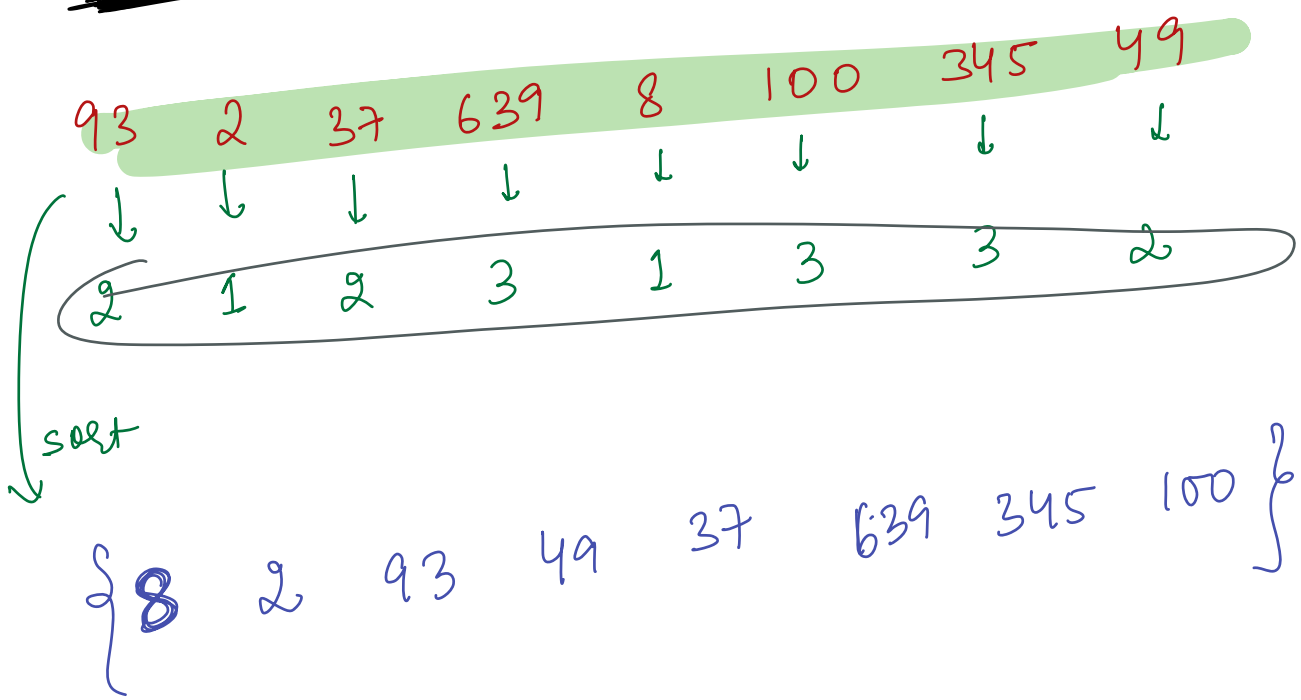
// print array
// print a & b

```
sort(ar[], comp);
```

takes 2 parameters and always returns a boolean value.

check how to use compare in your own language

Q: Given N array ele, sort in increasing order of no. of digits.
If two elements have same digits, ele with more value should come first



sort(_ , comp) → check in your respective language

true false

```
bool comp (int a, int b) {
    int d1 = digits(a)
    int d2 = digits(b)
    if (d1 < d2) return true
    else if (d1 == d2 && a > b) return true
    else return false
}
```

TC → O(n log n)