

Agenda

1. What & Why
2. Problems
3. Comparators/Keys

Not covered today - Sorting Algos (how they work)

Adv. DSA

Problem Solving Session
↓ ↓
5 - 7 , optional .

Sorting . → Arranging things in some order.

what can you sort

↳ numbers, strings, people, algorithms,
 ↓
 anything .

In what order you can sort ?

Increasing / Ascending .

$$A[i] < A[i+1]$$

[-10, 3, 6, 8, 12, 0 - 1]

→ [-10, -1, 0, 3, 6, 8, 12]

Decreasing / Descending .

$$A[i] > A[i+1]$$

→ [12, 8, 6, 3, 0, -1, -10]

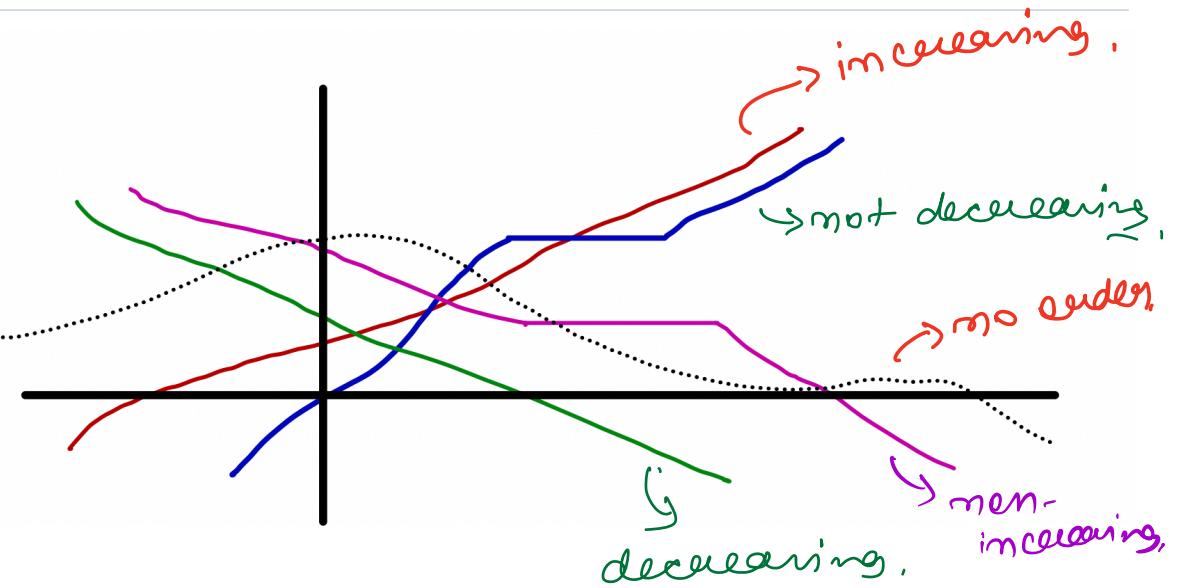
Non-increasing, $A[i] \geq A[i+1]$

$$[-5, 3, 6, 0, 6, 2, 0, 3]$$

$$\rightarrow [6, 6, 3, 3, 2, 0, 0, -5]$$

Non-Decreasing, $A[i] \leq A[i+1]$

$$[-5 \ 0 \ 0 \ 2 \ 3 \ 3 \ 6 \ 6]$$



Sort Based on what?

on Value [3, 10, 0, -6, 8, 5]

asc value

[-6, 0, 3, 5, 8, 10]

lexicographic (dictionary order)

['apple', 'deg', 'god', 'Scaler',
'algorithm', 'Smoke', 'algo']

↓ lexicographic asc.

['algo', 'algorithm', 'apple', 'deg', 'god', 'Scaler',
'Smoke']

By key (arbitrary property).

→ Sort list of people by height.

Dman , Sar , Ankush , Deepanshi , Shikhar
5'7 5'11 6 6'2 5'10

→ Dman, Shikhar, Sar, Ankush, Deepanshi -

→ Sort list of strings by length.

Jisha, Priyanka, Abhishek, Mangesha, Kaushal.
5 8 9 7 7

→ Jisha, Mangesha, Kaushal, Priyanka, Abhishek.

→ Sort list of Integers by absolute value.

[-10, 5, 6, 10, 12, -36, -20, 3]
↓ ↓ ↓ ↓ ↓ ↓ ↓
10 5 6 10 12 36 20 3

→ [3, 5, 6, -10, 10, 12, -20, -36].



stable sorting:-

→ [5, 6, 99, 43, 10, 20, 10]



[5, 6, 10, 10, 20, 43, 99].

Popular Sorting Algorithms

Bubble Sort

Quicksort

Radix Sort

Tim Sort

Selection Sort

Mergesort

Counting Sort

Shell Sort

Insertion Sort

Heap Sort

Bucket Sort

... ad infinitum

for the "Best" sorting algorithm:-

Time

Best Case $O(n)$

Avg Case $O(n \log n)$

worst Case $O(n \log n)$

Space :-

worst Case $O(1) / O(n)$

Stability →

If two elements compare same and their relative order, before and after sorting is same, the sorting algorithm is called stable.

Sorting a collection

Python (Tim Sort)	C++ (Quicksort + Insertion Sort)	Java (Tim Sort)
<pre>A = [...] A.sort() # modify the list. B = sorted(A) A = A.sort() # bad ↓ returns None.</pre>	<pre>vector<int> A; sort(A.begin(), A.end());</pre>	<pre>List<Integer> A; Collections.sort(A);</pre>

Ques)

Q1 - Elements Removal

Given integer array A.

Find the minimum cost of removing all elements from the array.

Cost of removing element A[i] = Sum of all elements left in the array before removing A[i]

A = [2, 1, 4]

$$\begin{array}{l} \downarrow \downarrow \downarrow \\ * \cancel{2}, \cancel{1}, 4 \\ \rightarrow 2 \rightarrow 7 \\ \rightarrow 1 \rightarrow 5 \\ \rightarrow 4 \rightarrow 4 \\ \hline 16 \end{array}$$

$$\begin{array}{l} \downarrow \downarrow \downarrow \\ * 2, 1, 4 \\ \rightarrow 1 \rightarrow 7 \\ \rightarrow 4 \rightarrow 6 \\ \rightarrow 2 \rightarrow 2 \\ \hline 15 \end{array}$$

3! \Rightarrow 5.

Ques $[4, 6, 1] \rightarrow 3! \Rightarrow 6$ ways.

$$4, 6, 1 \rightarrow 19$$

$$4, 1, 6 \rightarrow 24$$

$$6, 1, 4 \rightarrow 20$$

$$6, 4, 1 \rightarrow 12$$

$$1, 4, 6 \rightarrow 27$$

$$1, 6, 4 \rightarrow 25$$

$$\cancel{6} \cancel{1} \cancel{4} \rightarrow (6+1+4) + (1+4) + 4$$

$\rightarrow \underline{\underline{20}}$.

Brute force :-

try all possible arrangements.

$n!$ \rightarrow Bad Complexity.

Observations :-

1) If $A[i]$ is removed at step k , cost associated with $A[i] \rightarrow (A[i] + k)$.

Thus, later an element is removed, the more the cost associated with it.

2) we should remove elements in descending order.

$$A = [a, b, c, d, e]$$

remove	cost	remaining array.
a	$a+b+c+d+e$	b, c, d, e
b	$b+c+d+e$	c, d, e
c	$c+d+e$	d, e
d	$d+e$	e
e	e	-

$$1*a + 2*b + 3*c + 4*d + 5*e$$

Approach:-

1) Sort descending.

2) Suppose the array is [abcde],
between $(a+1 + b+2 + c+3 + d+4 + e+5)$.

```
public int min_cost (int [] arr) {
```

```
    // Sort arr in desc
```

```
    int total = 0;
```

```
    for (int i=0; i< arr.length; i++) {
```

```
        total = total + (arr[i] * (i+1))
```

3
return total .
3

T.C \rightarrow $O(n \log n)$

S.C \rightarrow $O(n)$.

Break

10:10 \rightarrow 10:18 PM,

HR and manager after seeing my
skill and work efficiency :-



iss lalu lal pe toh bina matlab
hi paise waste kar rahe hai

Ques)

Q2 - Noble Integers

Given N elements, count the number of noble integers present.

Noble Integer: x is noble \Leftrightarrow number of elements less than x is equal to x

$$A = \{ \frac{2}{1}, \frac{-1}{-5}, \frac{3}{3}, \frac{5}{5}, \frac{5}{5}, \frac{0}{-10}, \frac{4}{4} \}$$

$$A : [-10, \frac{0}{1}, \frac{1}{1}, \frac{3}{3}, 10]$$

$$A : [-3, \frac{0}{0}, \frac{2}{2}, 5]$$

$$A : [-10, -5, 1, 3, 4, 5, 10]$$

$$A : [-3, \frac{0}{0}, \frac{1}{2}, \frac{2}{2}, \frac{4}{5}, \frac{4}{5}, \frac{4}{5}, \frac{4}{5}, \frac{8}{8}, \frac{8}{8}, \frac{10}{10}, \frac{10}{10}, \frac{10}{10}, 13]$$

$\Rightarrow 7$.

-: Brute Force :-

```
Public int Noble (int[] arr, int n) {
```

```
    noble = 0;
```

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

```
        count = 0;
```

$$1. C \Rightarrow O(n^2)$$
$$2. C \Rightarrow O(\underline{n})$$

```
for (j = 0; j < n; j++) {  
    if (arr[j] < arr[r]) {  
        count++;  
    }  
    if (count == arr[r]) {  
        mable++;  
    }  
}
```

// Assume no repetition.

Observations :-

- 1) :- — we no' can never be noble,
 - 2) g f no duplicates :-

$$A = [10, 1, 8, -5, 4, -10]$$

↓ ↓ ↓ ↓ ↓ ↓ ↓
 5 2 3 -1 4 0

\downarrow

$[-10, -5, 1, 3, 4, 10]$

0 1 2 3 7 9

Approach :-

1) Sort Ascending.

2) if $A[i] == i$, then increment noble count.

* Duplicacy Scenario

→ [-10, 1, 1, 2, 4, 4, 4, 8, 10]
 0 1 2 3 4 5 6 7 8
 ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
 0 1 1 3 4 4 4 7 8

public int count_nobles (int[] A)

int[] arr = sort (A)

int lesser = 0

int noble = 0;

for (i=0, i < A.length, i++) {

if (i > 0) && arr[i] == arr[i-1])

{ // repetition part .

3 else {

 lesser = i

}

if ($\text{arr}[i] == \text{target}$) {

$\text{mable} += 1$

 3
 3 [-10, 1, 1, 2, 4, 4, 4, 8, 10]
 0 1 2 3 4 5 6 7 8

 2
 i target mable
 0 0
 1 1
 2 1

11:02 → 11:04

← Custom Sorting ↳

49, 16, 28, 30, 10, 5
↓ ↓ ↓ ↓ ↓ ↓
3 5 3 8 4 2

Sent them

bored on

their no'

of factors.

1
2
3
4
5
6
7
8
9
10
11
12

$\rightarrow \{5, 49, 25, 10, 16, 30\}$

Collections Jew (Arr).

* Wishal, Anay, Abhishek, Sai, Venkata.
 ↓ ↓ ↓ ↓ ↓
 6 4 2 3 7.

* Comparators :->

Comparators in C++

```
bool compare(int a, int b) {
    /*
        if a should appear before b then return true
        else return false
        Caution: if a and b compare equal, then return false
    */
}
```

$\text{arr} \rightarrow 49, 16, 25, 30, 10, 5 \rightarrow$ sort based
 ↓ ↓ ↓ ↓ ↓ ↓ on no. of
 3 5 3 8 4 2 factory.

$\rightarrow \text{sort}(\text{arr.begin()}, \text{arr.end()}, \text{compare})$

```
bool compare (int a, int b) {
    int fact1 = factors(a)
    int fact2 = factors(b)
    if (fact1 < fact2) { return true }
    else { return false }
```

3

Comparators in C++

```
bool compare(int a, int b) {
    /*
        if a should appear before b then return true
        else return false
        Caution: if a and b compare equal, then return false
    */
}
sort(v.begin(), v.end(), compare)
```

Comparators in Java

```
Comparator customComparator = (Integer a, Integer b) -> {
    /*
        if a should appear before b, then return -1
        if a and b compare equal, then return 0
        else, return 1
    */
}

Collections.sort(items, customComparator)
```

↓
away now .

Comparators in Python (similar to Java)

```
from functools import cmp_to_key

def compare(a, b):
    """
        if a should appear before b, then return -1
        elif a and b compare equal, then return 0
        else, return 1
    """

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

Arrays.sort (Au)



Comparator To