Recap: ① Sorting → Mergesort and Quicksort → $O(n \log_2 n)$

Solve sorting based questions using in-built sort() function.

sort() C++/Java → Timsort. ← sort() Py

Insertion sort for small n $n < 100$ $O(n^2)$

quicksort $n > 100$ $O(n \log n)$

Sorting

NUMBERS TREATED LIKE STRINGS.

Q.1 arr = [18, 6, 78, 81, 9, 98, 23, 45]

Arrange these numbers in such a way that when concatenated it forms the largest number.

Eg: [9, 98, 81, 78, 6, 45, 23, 18] ⇒ 998178645231 8

Sol: Sorting in general sort(arr) → [6, 9, 18, 23, 45, 78, 81, 98]

strings = "sea" "mountain" "sight" L → R

Imagine: These elements are strings and sort them with that comparison logic. Sort has to be in descending order.

[1 2 1 4 9 3 1 5 2 8]

Q.2 str = "I am a data scientist and I teach at HeyCoach"

Sort the words in the sentence based on length. Tie resolution: The word that appears first in input, also appears first in o/p.

o/p: "I a I am at and data teach HeyCoach scientist"

Sol: Associate each word to its length & sort the lengths. Corresponding strings will get sorted as well.

STRINGS TREATED AS NUMBERS

Q.3 arr = ["apple", "google", "strawberry", "peach",
"maple", "gooseberry", "blackberry", "cherry",
"reach", "toggle", "staple"].

THINK SIMPLE

STRINGS TREATED AS ANOTHER STRINGS

Arrange them such that rhymes are clustered together.

"elppa" "elpam" "elpats" "elggot" ...

o/p : ["apple", "maple", "staple", "toggle", "google", "peach",
"reach", "cherry", "blackberry", "gooseberry", "strawberry"]

Sol: apple maple → elppa elpam elpats

Imp note : Do not touch your original array !!

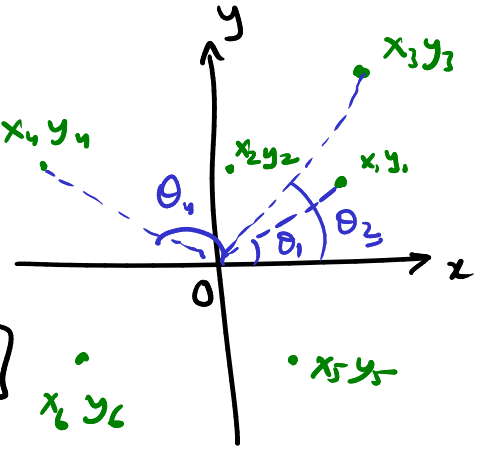
OBJECTS TREATED AS NUMBERS

Q.4 arr = $[(x_1, y_1), (x_2, y_2), (x_3, y_3), \cdots, (x_n, y_n)]$

Codeforces contest #1

points on 2D plane

$\left[\frac{y_1}{x_1} \quad \frac{y_2}{x_2} \quad \frac{y_3}{x_3} \quad \cdots \frac{y_n}{x_n}\right]$



Sort these points smaller the angle, earlier it appears.

o/p : $[x_1 y_1, x_3 y_3, x_2 y_2, x_4 y_4, x_6 y_6, x_5 y_5]$

Sol: ① Sorting sort() f^n is not defined for custom object.

Associate objects to numbers & then correspondingly sort those objects.

15 lakh

HW. Sort JSON objects.

JEE Rank is decided

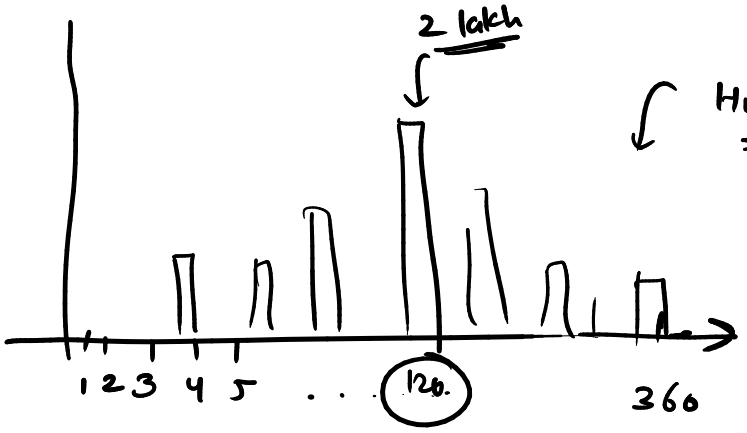
```
[ { name : "Saga"
    dob  : "05 Aug -"
    phy  : 100   /120
    chem : 80    /120
    maths: 95    /120
```

cond^n
① Total marks ↑ if tie.
② Maths ↑ if tie
③ Phy ↑ if tie
④ Chem ↑ if tie
④ age ↓

} , { }].

JEE → 15 lakh students. 0 - 360 | assume no marks < 0.

How will you determine the rank ?



Idea: $[a_1 \; a_2 \; a_3 \; a_4 \; \ldots \; a_n] \Rightarrow$ an.

values

Ordering

Coupling.

$[f(a_1) \; f(a_2) \; f(a_3) \; f(a_4) \; \ldots \; f(a_n)] \Rightarrow$ key array

Associate

Swap

Sort the keys.

Sort this array.

an → order you want !!

Generalized look at sorting → ordering | some pattern. underlying sorted.

C++ / Java

an = [0.0.0.0]
↑ type type

Sort (start of the array , end of the array , compare f^n.)

a to appear before b.

```
bool compare (type a, type b) {
    return f(a) < f(b);
}
```

more preferred Py

an. Sort (key = f)

ai name

lambda expression

an.sort (key = (a,b) ⇒ (f(a) - f(b)))

$O(n \log n)$ ⟹ Comparison-based sort.

2 elements a_i a_j → are they in correct order

if not ⟹ swap it !!

Sort f^n: ⟹ Insertion sort $n < 100$ quick sort. $n > 100$

Ordering in qs → partition f^n: → if cond^n: swap(-,-)

* Job : For all pairs considered by qs to compare, it uses the compare f^n:

If the comparison returns true ⟹ order is correct

If the comparison returns false ⟹ sort algo performs swap.

arr = [3 5 4 6] Sort. a < b

a b

pairs: (3, 5) ⟹ True ✓

(5, 4) ⟹ False → a b [3 4 5 6]

(5, 6) ⟹ True ✓