AUGUST 23, 2025

DOUBT SESSION

Ques 1. Maximise the fraction (BinSearch / Application Idea 3)

Given two arrays A and B of size N and an integer K. You have to select K indexes $i_1, i_2, i_3, \ldots, i_K$ such that

$$\frac{A[i_1] + A[i_2] + A[i_3] + \cdots + A[i_K]}{B[i_1] + B[i_2] + B[i_3] + \cdots + B[i_K]}$$

is maximum.

When is a fraction maximum?

frac (max) = num ↑ (increasing) / deno ↓ (decreasing)

Example : 10 9 7
3 5 4

K=2 (can't have more than 2 pairs in pq)

- sorted pairs

(10|3) (9|5) (7|4)
↓ ↓ ↓
3.33 1.8 1.75

- priority queue of pairs

numerator sum = 0 (num)
denominator sum = 0 (den)

Priority queue can have atmost 2 elements.

pq.size() = 0 < 2
pq.insert(10|3)
num += 10 = 10
den += 3 = 3
pq.size() = 1 < 2
pq.insert(9|5)

num += 9 = 19
den += 5 = 8
pq.size() = 2 (Full)

pq
10,3
9,5

Current ratio = 19|8 = 2.375
Min ratio pair in pq = 9|5
If we want to include the pair (7|4) then we first have to remove 9|5

num -= 9 = 10
den -= 5 = 3

pq
10,3
~~9,5~~
7,4

pq.insert(7|4)
pq.size() = 2 (Full)
num += 7 = 17
num += 4 = 7
New ratio = 17|7 = 2.428
New ratio > Current ratio (Better choice)

Ans = New Ratio = 2.428

Example : 4 4 2 1 5 3 2 5
2 2 2 5 3 5 2 3

K = 3 (no more than 3 pairs in pq)

- sort below pairs before insert in pq

(4|2) (4|2) (2|2) (1|5) (5|3) (3|5) (2|2) (5|3)
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
2 2 1 0.2 1.67 0.6 1 1.67

numerator sum, num = 0
denominator sum, den = 0
pq.size() = 0 < 3
pq.insert(4,2)

AlgoZenith

num += 4 = 4
den += 2 = 2
pq.size() = 1 < 3
pq.insert(4,2)
num += 4 = 8
den += 2 = 4
pq.size() = 2 < 3
pq.insert(5,3)
num += 5 = 13
den += 3 = 7

pq

4,2
4,2
~~5,3~~

pq.size() = 3 < 3 (Full)
Current Ratio = 13/7 = 1.857
pq.pop() [Rem 5,3]
num -= 5 = 8
den -= 3 = 4
pq.size() = 2 < 3
pq.insert(5,3)
num += 5 = 13
den += 3 = 7

pq

4,2
4,2
5,3

New Ratio = 13/7 = 1.857
Current Ratio = New Ratio
pq.size() = 3 < 3 (Full)
pq.pop [Rem 5,3]
num -= 5 = 8
den -= 3 = 4
pq.size() = 2 < 3
pq.insert(2,2)
num += 2 = 10
den += 2 = 6

pq

4,2
4,2
2,2

New ratio = 10/6 = 1.666
Current Ratio > New Ratio
pq.size() = 3 < 3 (Full)
pq.pop() [Rem (2,2)]
Again the (2,2) will insert and rem like above.
Then the next insertion will be of the rest of the pair. But Evidently Ans = Current Ratio = 1.875

custom sort of pairs from highest to lowest ratios.
Maintain K sized priority queue with highest ratio pair & numerator sum, denominator sum
Maintain 2 Answer ratio comparator

Ques 2. Code for 1

https://codeforces.com/problemset/problem/768/B

Jon fought bravely to rescue the wildlings who were attacked by the white-walkers at Hardhome. On his arrival, Sam tells him that he wants to go to Oldtown to train at the Citadel to become a maester, so he can return and take the deceased Aemon's place as maester of Castle Black. Jon agrees to Sam's proposal and Sam sets off his journey to the Citadel. However becoming a trainee at the Citadel is not a cakewalk and hence the maesters at the Citadel gave Sam a problem to test his eligibility.

Initially Sam has a list with a single element n. Then he has to perform certain operations on this list. In each operation Sam must remove any element x, such that $x > 1$, from the list and insert at the same position $\lfloor \frac{x}{2} \rfloor, x \mod 2, \lfloor \frac{x}{2} \rfloor$ sequentially. He must continue with these operations until all the elements in the list are either 0 or 1.

Now the masters want the total number of 1s in the range l to r (1-indexed). Sam wants to become a maester but unfortunately he cannot solve this problem. Can you help Sam to pass the eligibility test?

Mid element = n%2

0/1

n/2] n%2 [n/2

For 10

10%2 = 0

10/2 = 5

don't do this function call

Just calc the L·H·S & use it on R·H·S for this

Ans to query

1011101011101

5] 0 [5

Breaking down this would be sufficient and be used

Return to parent

For 5

5%2 = 1

5/2 = 2

= 101 1 101

2] 1 [2 ← use it here as well

Break this down

Return to parent

For 2

2%2 = 0

2/2 = 1

1] 0 [1

= 1 0 1

Return to parent

For 1

1%2 = 1

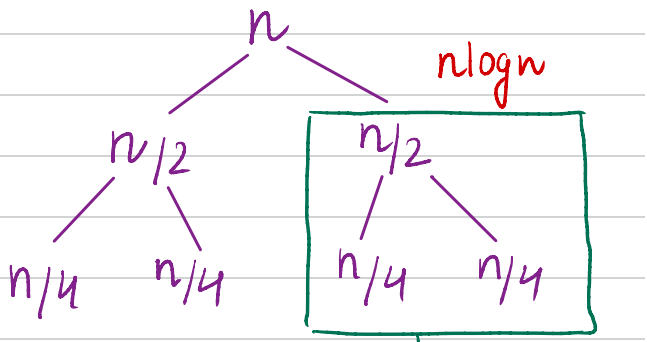
1/2 = 0

1

We can query L to R on Ans to find no. of 1's.

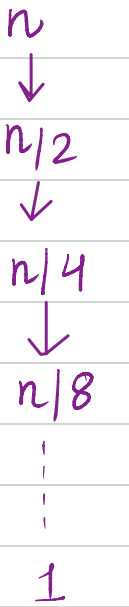
Use strings, not vectors

- Brute Idea:



No need to repeat the right tree.

Our solution approach:



O(log N)

Ques 3. Cellular Network

https://codeforces.com/problemset/problem/702/C

You are given n points on the straight line — the positions (x-coordinates) of the cities and m points on the same line — the positions (x-coordinates) of the cellular towers. All towers work in the same way — they provide cellular network for all cities, which are located at the distance which is no more than r from this tower.

Your task is to find minimal r that each city has been provided by cellular network, i.e. for each city there is at least one cellular tower at the distance which is no more than r.

If $r = 0$ then a tower provides cellular network only for the point where it is located. One tower can provide cellular network for any number of cities, but all these cities must be at the distance which is no more than r from this tower.

Binary Search on Ans

- Ans range

Max. Range of Tower = $|c[0] - c[n-1]|$
= abs(c[0] - c[n-1])

leftmost city ↑ , rightmost city ↑

Min. = 0

0 ⟷ abs(c[0] - c[n-1])

Example : -3 . 2 4

Min = 0

Max = 4 - (-3) = 7

Mid = $\frac{0+7}{2}$ = 3

Range: [-6 - - - 3]

Only 4 is outside the range.

Increase the range to include 4.

- If all cities got covered
Minimise the range
mid → r → will keep this as probable answer as it covers all cities

Search 0 _ _ _ mid-1 for better answer Update ans if better range is found.

Store ranges in set/map.

To see if a city is covered within the range.

Based on value of r, perform Binary Search.