# Doubt Session

## 1. Maximize the fraction : Binary Search

Approach :-

1. Sort : Order all pairs by their individual A/B ratio, from high to low.
2. Fill a priority queue of size k with the top k pairs to get an initial solution.
3. For each remaining pair, test if swapping it with any of the pair currently in the queue yields a better total ratio. If it does, perform the swap and update your best ans.

Ex: 10 9 7
3 5 4

① Pair by Ratio : (10, 3) → 3.33
(9, 5) → 1.8
(7, 4) → 1.75

② A priority que (pq) of size k=2 is filled with first 2 pairs from the sorted list.

Add (10, 3) :
num_sum = 10, den_sum = 3

| 10,3 |
|---|
| 9,5 |

Add (9,5) :
num_sum = 10+9 = 19, den_sum = 3+5 = 8

Priority queue is now filled with [(10,3), (9,5)]

③ Current Ratio : 19/8 = 2.375

④ Now consider the next pair (7,4) and check if swapping it with a pair already in the queue improves the ratio. Replace with lowest value in the current set i.e, (9,5).

→ Remove (9,5) and add (7,4)

→ New sum : new num_sum = (19-9) + 7 = 17
new den_sum = (8-5) + 4 = 7

→ New Ratio : $\frac{17}{7}$ = 2.428

5. Compare and : 2.428 (New Ratio) > 2.375 (current ratio).

So, new pairs are (10,3) and (7,4) and max. achievable ratio is 2.428.

2. Codeforces [768 B]

middle element = n%2

n/2] n%2 [n/2]

↓

0/1

Eg:- 10

10 % 2 = 0 , 10/2 = 5 ⇒ 5] 0 [5 ✗ (not required)

↓

Break down this

5 % 2 = 1, 5/2 = 2 ⇒ ✓2] 1 [2 ✗

2 % 2 = 0, 2/2 = 1 ⇒ 1] 0 [1 ⇒ [1 0 1]

Now, 5 → 101 1 101

Now 10 → 1011101 0 1011101 (L, R)

We can query L to R, to find overall 1's present there.

Don't form vectors, use strings.

3. Cellular Network - Codeforces [702 C]

↳ Binary search on Ans

~~Instead of checking radii one by one find the best radius by guessing.~~

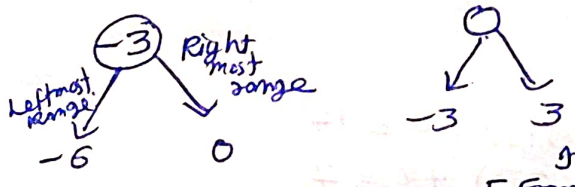
Max. range of tower = |c[0] - c[n-1]|

= abs(c[0] - c[n-1]).

Eg: -3 2 4

Min = 0 Mid = 3

Max = 7




-6 ---- 3 [In this range, ~~from here~~, any tower t can provide service]

4 is outside range, so if any of the city left we have to inc. the range.

Minimise the range: store mid → ans (it can be a probable ans).

Search in 0 ---- mid-1 for better ans. Update if found better.