

Day - 192Heap - 2

* Smallest Range covering all elements from k Lists:

⇒

4	10	15	24	26
0	9	12	20	
5	18	22	30	

⇒

We have to return a range in which atleast ~~all~~ one element from all the list should come.

Ex: $[0, 5] \rightarrow \text{difference} = 5$

⇒

But, we have to return smallest range —
i.e. $[20, 24] \rightarrow \text{difference} = 4$

⇒

We can make a single array then make a range & check if that range is valid or not.

⇒

After that, we will return the smallest range.

⇒

This is the brute force method.

T.C. $\rightarrow O(nk)^3$

⇒

We will use min-heap, here —

⇒

So, we will select the min element of all the list & make a range with min & max from that.

Ex: $4, 0, 5 \rightarrow [0, 5]$

Date _____

Page _____

⇒ Now, we have to optimize the range, for that we can increase the smallest no. from range or decrease the largest.
 ⇒ But we will select only increase case.

⇒ Now, we will pop the smallest from the no. & push the next of that list & make new range.

Ex: 4, 0, 5 → 4, 9, 5 →
 [4, 9]

⇒ And when we get smallest range, we will update the answer.

* Construct Target array with multiple sum:

9	3	5
---	---	---

⇒ We have to return the target array by using

1	1	1
---	---	---

.

⇒ We can sum of all the element then write that answer in any index of the array.

⇒ If we make the target array by using this approach, we will return 1.

1	1	1	→ 3 →	1	3	5
1	3	5	→ 5 →	1	3	5
1	3	5	→ 9 →	9	3	5

Sum

Target array

⇒ we can't directly update any index of the array. ⇒

⇒ So, we will do the reverse process.

⇒ we will start from a target array.

⇒ Then select the max. element & subtract it from the sum of target array. ⇒

⇒ Now, we have subtract that result by max. element.

⇒ Then that answer, we will write on the max. element place.

⇒

10	25	1	37
----	----	---	----

 → 37

$$\text{Sum} = 73$$

$$73 - 37 = 36$$

$$37 - 36 = 1 \rightarrow \text{ans}$$

⇒ If ~~For~~ $\text{RemSum} \leq 0$ or $\text{RemSum} \geq \text{max}$ return 0.

⇒ Our this logic will give TLE.

⇒ Ex:

1	20
---	----

1	19
---	----

1	18
---	----

⋮

1	1
---	---

⇒ So, we only updating one element.
So, if in place of 20, if we have

Date _____

Page _____

=> more larger no. then, it will take more time.
So, we have to do this in one step.

Element = $\text{MaxEle} \% \text{RemSum}$;

=> * we can also get 0, so, if we get 0 & remSum is 1 then only answer is possible otherwise not.

> max