Array : Video - 84

Leetcode - 2402

~~Hard~~ → Easy

Facebook
Instagram → codestorywithMIK

(Twitter) → CSwithMIK

codestorywithMIK →

Google

R0 R1 R2

m1 (4, 10) m2 (1, 6)

2402. Meeting Rooms III

Hard | Topics | Companies | Hint

You are given an integer `n`. There are `n` rooms numbered from `0` to `n - 1`.

You are given a 2D integer array `meetings` where `meetings[i] = [start_i, end_i]` means that a meeting will be held during the **half-closed** time interval `[start_i, end_i)`. All the values of `start_i` are **unique**.

Meetings are allocated to rooms in the following manner:

1. Each meeting will take place in the unused room with the **lowest** number.
2. If there are no available rooms, the meeting will be delayed until a room becomes free. The delayed meeting should have the **same** duration as the original meeting.
3. When a room becomes unused, meetings that have an earlier original **start** time should be given the room.

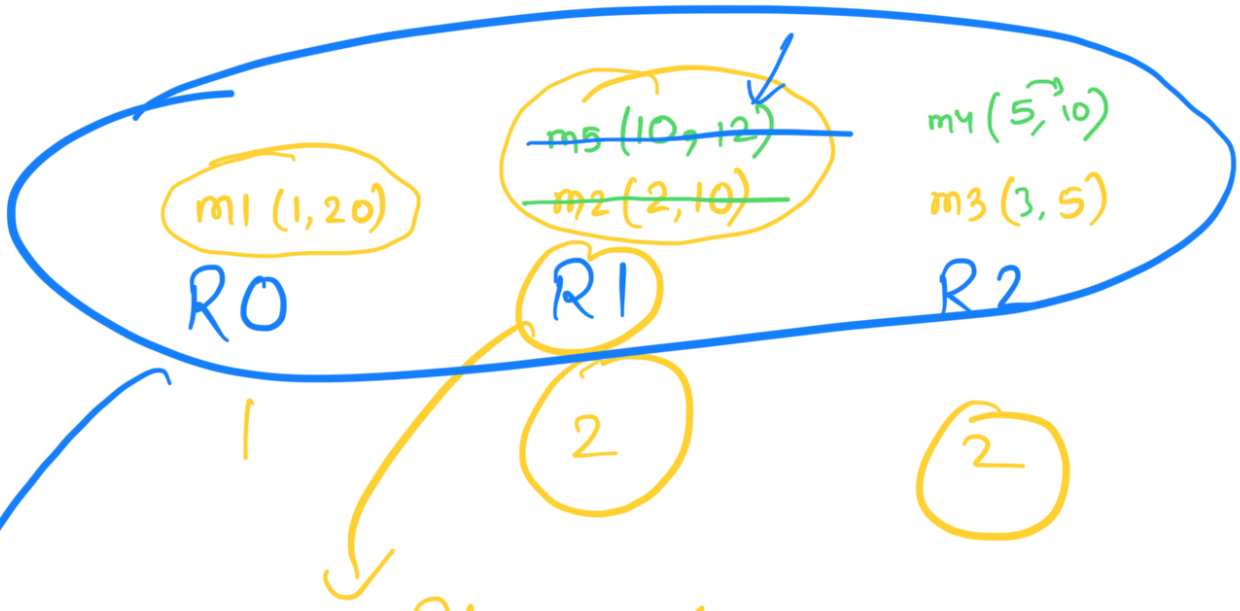
Return *the **number** of the room that held the most meetings.* If there are multiple rooms, return *the room with the **lowest** number.*

Sort based on start-time.

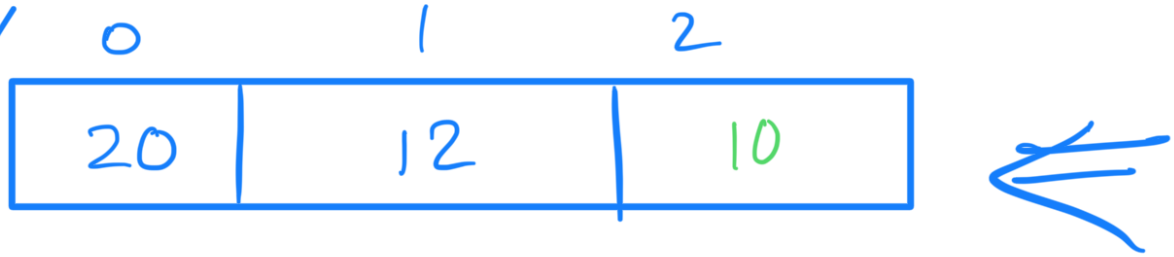
Example :- $n = 3$

meetings = [m1 (1, 20), m2 (2, 10), m3 (3, 5), m4 (4, 9), m5 (6, 8)]

(8 - 6 = 2)



R1 = 1.



Conclusion:-

① Sort meetings (start_time).

② Find the first room which can be used for current meeting.

↳ Empty Room found. ⇐

↳ No Room found. ⇐

↳ Select the room which will end first.

Approach-2

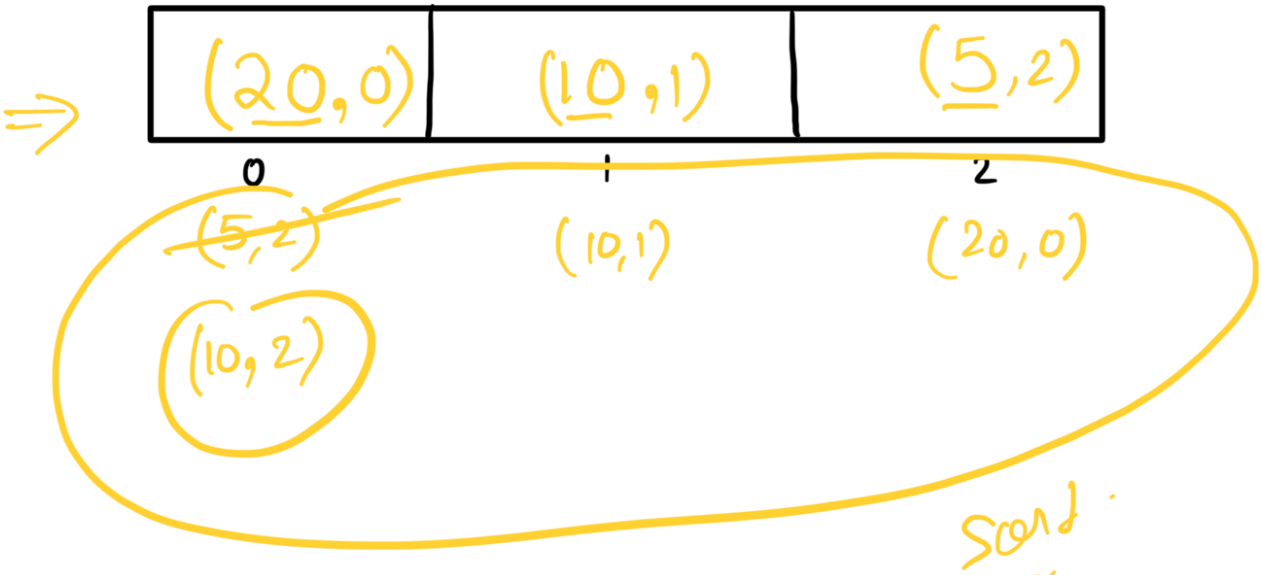
what was the bottleneck in the first approach ???

O(m)

m1 m2 m3 m4 m5

[ ]

meetings = [(1, 20), (2,10), (3,5), (4,4), (6,8)]



Sort.
↳ Earliest ending time.

$O(m * n\log n)$

Heap (min-heap).
top

meetings = [m1 (1, 20), m2 (2, 10), m3 (3, 5), m4 (4, 9), m5 (6, 8), (13, 20)]

$$T.C = O(m * \log(n))$$

$(m * n)$