# Meeting Rooms III – Problem Breakdown and Solution

## 📘 Problem Summary

You're given an integer n (number of meeting rooms numbered from 0 to n-1) and a list of meetings, where each meeting is given as [start, end]. Your task is to simulate the room assignments and return the room that held the most meetings.  
  
Key constraints:  
- Assign meetings to the lowest numbered free room.  
- If all rooms are busy, delay the meeting until one becomes free (keeping the same duration).  
- When assigning delayed meetings, original start time is used to choose the lowest numbered room.  
- Return the room with the most meetings, using smallest number in case of tie.

## 🪜 Step-by-Step Algorithm Design

1. Sort the meetings by start time.

2. Use two heaps:  
 - One for available room numbers (min-heap)  
 - One for busy rooms as (end\_time, room\_number)

3. For each meeting:  
 - Free up any rooms where meetings have ended by the current meeting's start.  
 - If a room is available, assign it.  
 - If not, delay the meeting to the earliest available room's end time.  
 - Always choose the lowest-numbered room.

4. Track how many meetings each room hosted.

5. Return the room with the highest count (lowest index if tie).

## 🧪 Full Python Code

