

# Meeting Rooms II

Try to solve the Meeting Rooms II problem.

We'll cover the following ^

- Statement
- Examples
- Understand the problem
- Figure it out!
- Try it yourself

## Statement

We are given an input array of meeting time intervals, `intervals`, where each interval has a start time and an end time. Your task is to find the minimum number of meeting rooms required to hold these meetings.

Constraints:

- $1 \leq \text{intervals.length} \leq 10^4$
- $0 \leq \text{start}_i < \text{end}_i \leq 10^6$

## Examples

Sample example 1

Input

Intervals	[2, 8]	[3, 4]	[3, 9]	[5, 11]	[8, 20]	[11, 15]
-----------	--------	--------	--------	---------	---------	----------

Output

Meeting room 1

Interval 1 [2, 8]
Interval 5 [8, 20]

Meeting room 2

Interval 2 [3, 4]
Interval 4 [5, 11]

Meeting room 3

Interval 3 [3, 9]
Interval 6 [11, 15]

A total of 3 meeting rooms are required to hold these 6 meetings.

1 of 3

## Understand the problem

Let's take a moment to make sure you've correctly understood the problem. The quiz below helps you check if you're solving the correct problem:

1

Meeting time intervals = [ [1, 3], [2, 6], [8, 10], [9, 15], [12, 14] ]

How many meeting rooms are required to hold these meetings?

A) 3

B) 1

C) 2

Submit Answer



Question 1 of 3  
0 attempted



Reset Quiz ↺

## Figure it out!

We have a game for you to play. Rearrange the logical building blocks to develop a clearer understanding of how to solve this problem.

**Note:** As an additional challenge, we have intentionally hidden the solution to this puzzle.



Drag and drop the cards to rearrange them in the correct sequence.

Loop over the remaining intervals.

Initialize a min-heap and push the end time of the first interval onto it.

Otherwise, allot a new meeting room, that is, add the current interval in the heap without removing any existing interval.

Sort the given input intervals with respect to their start times.



In each iteration, compare the start time of the current interval with all the end times present in the heap.

After processing all the intervals, the size of the heap is the count of meeting rooms needed to hold the meetings.

If the earliest end time of all intervals seen so far (the root of the heap) occurs before the start time of the current interval, remove the earliest interval from the heap and push the current interval onto the heap.

Reset

Submit

## Try it yourself

Implement your solution in `MeetingRooms.java` in the following coding playground. You will need the provided supporting code to implement your solution.

**Note:** We have left the solution to this challenge as an exercise for you. You may try to translate the



Java

MeetingRooms.java

Interval.java

```
1 import java.util.*;
2
3 public class MeetingRooms{
4     public static int minMeetingRooms(List <Interval> intervals) {
5
6         // Your code will replace this placeholder return statement
7         return -1;
8     }
9 }
```

Powered by AI



Submit

Test Cases Results

Case 1

Case 2

Case 3



Input #1

[[2,8],[3,4],[3,9],[5,11],[8,20],[11,15]]

Meeting Rooms II

← Back

Next →

Solution: Employee Fr...

In-place Reversal of a ...

☒ Mark as  
Completed