

- ChatGPT
- Explore GPTs
- Today
- Merge Intervals Example
- Fast and Slow Pointers
- Matrix Pattern Problems
- Two Pointers Problems
- Matrix Traversal Problems
- Sliding Window Patterns
- SQL Window Functions Explained
- ACID Properties Explained
- New chat
- SQL Query Performance Tips

Merge intervals pattern-based problem that are mostly asked 10 interview question
give input and output of problem.
add sample problem like conflicting appointments,
minimum meeting rooms

< 2/2 >

Sure! The "Merge Intervals" problem is a common pattern-based question that often comes up in interviews. It typically involves combining overlapping intervals or ranges.

Here's a brief overview of the problem along with a sample problem related to conflicting appointments and minimum meeting rooms:

Problem: Merge Intervals

Description:



Get citation

You are given a collection of intervals, where each interval is represented as a pair `[start, end]`. Your task is to merge all overlapping intervals and return a new list of non-overlapping intervals.

Input:

- A list of intervals where each interval is an array of two integers: `[start, end]`.

Output:

- A list of merged intervals, sorted by their starting times.

Example:

Input:

plaintext Copy code

`[[1, 3], [2, 6], [8, 10], [15, 18]]`

Output:



plaintext Copy code

`[[1, 6], [8, 10], [15, 18]]`

Explanation:

- The intervals `[1, 3]` and `[2, 6]` overlap and can be merged into `[1, 6]`.
- The intervals `[8, 10]` and `[15, 18]` do not overlap with any other intervals.

Sample Problems

1. Conflicting Appointments

Description:

Given a list of appointments, determine if any of them conflict. Each appointment is represented as an interval `[start, end]`. Two appointments conflict if their intervals overlap.

Input:

- A list of intervals where each interval represents an appointment.



Output:

- Return `true` if any appointments conflict, otherwise return `false`.

Example:

Input:

plaintext Copy code

```
[[1, 3], [2, 6], [8, 10], [15, 18]]
```

Output:

```
plaintext
```

[Copy code](#)

```
true
```

Explanation:

- The appointments `[1, 3]` and `[2, 6]` overlap, so there is a conflict.

2. Minimum Meeting Rooms

Description:

Given a list of intervals representing meeting times, determine the minimum number of meeting rooms required to accommodate all meetings without any overlap.

Input:

- A list of intervals where each interval represents a meeting time `[start, end]`.

Output:

- An integer representing the minimum number of meeting rooms required.

Example:

Input:

```
plaintext
```

[Copy code](#)

```
[[1, 3], [2, 6], [8, 10], [15, 18]]
```



Output:

```
plaintext
```

[Copy code](#)

```
2
```

Explanation:

- The meetings `[1, 3]` and `[2, 6]` overlap and require two meeting rooms to be scheduled simultaneously. All other meetings do not overlap with each other.

Approach

1. Merge Intervals:

- Sort the intervals based on the starting time.
- Iterate through the sorted intervals, merging overlapping ones.

2. Conflicting Appointments:



- Use the merging intervals approach and check if the merged result contains any interval with more than one overlap.

3. Minimum Meeting Rooms:

- Sort the intervals based on start times.
- Use a priority queue (min-heap) to track the end times of meetings.
- Iterate through the intervals, and for each one, check if a room can be freed or if a new room is needed.

Would you like code examples for any of these problems?



