Merge Intervals: Introduction

Let's go over the Merge Intervals pattern, its real-world applications and some problems we can solve with it.

We'll cover the following

- Overview
- Examples
- Does my problem match this pattern?
- Real-world problems
- Strategy time!

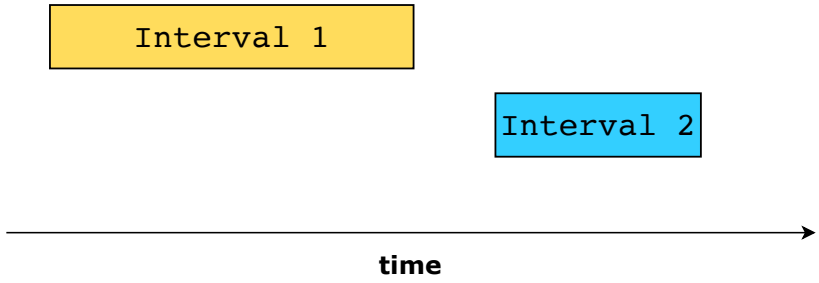
Overview

The **merge intervals** pattern deals with problems involving overlapping intervals. Each interval is represented by a start and an end time. For example, an interval of $[10, 20]$ seconds means that the interval starts at 10 seconds and ends at 20 seconds, such that both 10 and time 20 are included in the interval.

The most common problems solved using this pattern are scheduling problems.

The key to understanding this pattern and exploiting its power lies in understanding how any two intervals may overlap. The illustration below shows the six different ways in which two intervals can relate to each other:

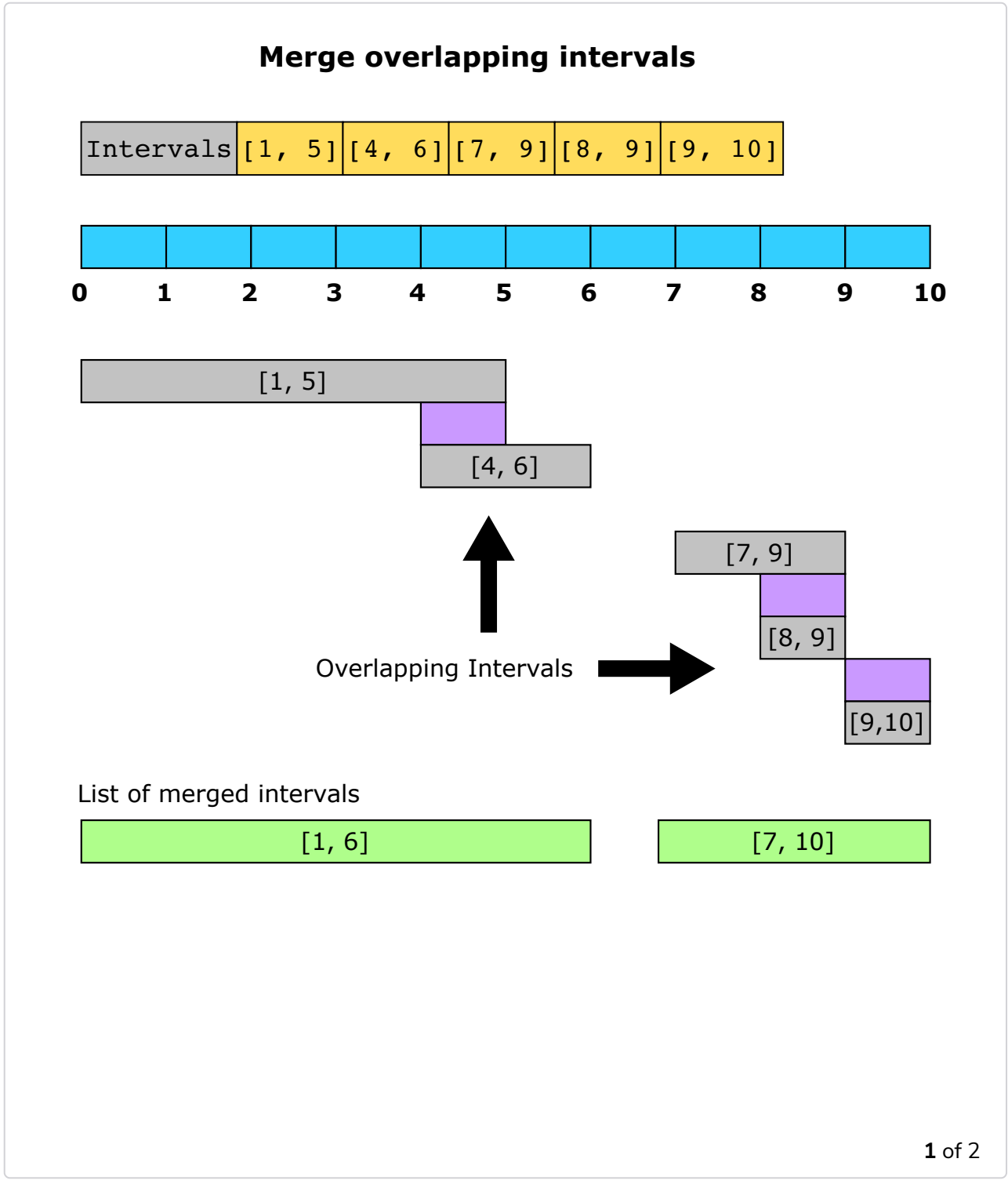
1. Intervals 1 and 2 don't overlap. Interval 1 ends before the start of Interval 2:



1 of 6

Examples

The following examples illustrate some problems that can be solved with this approach:



Does my problem match this pattern?

- Yes, if both of these conditions are fulfilled:
  - The input data is an array of intervals.
  - The problem requires dealing with overlapping intervals, either to find their intersection, their union, or the gaps between them.

This may be required as the final goal, or as an intermediate step

- The order of the intervals in the result is not significant.
- The input list of intervals is not sorted. In such a situation, we would prefer to use some other technique to efficiently solve the problem.

Real-world problems

Many problems in the real world use the merge intervals pattern. Let's look at some examples.

- **Display busy schedule:** Display the busy hours of a user to other users without revealing the individual meeting slots in a calendar.
- **Schedule a new meeting:** Add a new meeting to the tentative meeting schedule of a user in such a way that no two meetings overlap each other.
- **Task scheduling in operating systems (OS):** Schedule tasks for the OS based on task priority and the free slots in the machine's processing schedule.

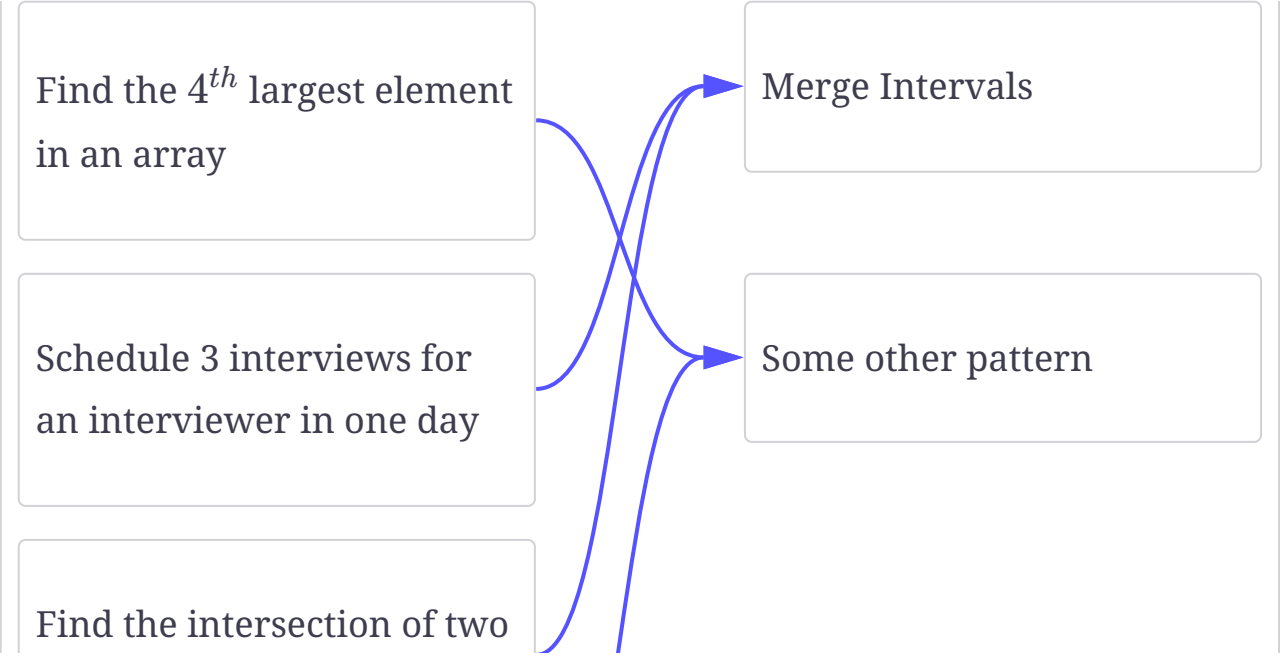
Strategy time!

Match the problems that can be solved using the merge intervals pattern.

> **Note**: Select a problem in the left-hand column by clicking it, and then click one of the two options in the right-hand column.

Match The Answer

Select an option from the left-hand side


Find the 4^{th} largest element in an array
Schedule 3 interviews for an interviewer in one day
Find the intersection of two
Merge Intervals
Some other pattern