

Insert Interval

Try to solve the Insert Interval problem.

We'll cover the following ^

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

Statement

Given a sorted list of **nonoverlapping** intervals and a new interval, your task is to insert the new interval into the correct position while ensuring that the resulting list of intervals remains sorted and nonoverlapping. Each interval is a pair of nonnegative numbers, the first being the start time and the second being the end time of the interval.

Constraints:

- $0 \leq \text{existing_intervals.length} \leq 10^4$
- $\text{existing_intervals}[i].\text{length}, \text{new_interval.length} == 2$
- $0 \leq \text{start time}, \text{end time} \leq 10^4$
- The first number should always be less than the second number in each interval.
- The list of intervals is sorted in ascending order based on the first element in every interval.

Examples

Sample example 1

Input

Existing intervals

[1, 3] [5, 7] [8, 9] [10, 13]

New intervals

[2, 6]

Output

[1, 7] [8, 9] [10, 13]

We will merge [2, 6] with the first interval, [1, 3], to create [1, 6] and then merge this interval with the next overlapping interval, [5, 7], to create [1, 7]. The intervals [8, 9] and [10, 13] don't overlap with any intervals, so they will exist independently.



Understand the problem

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

Insert Interval

1

What will be the updated list of intervals?

existing_intervals = [[1, 3], [4, 5], [7, 8], [9, 12], [13, 14]]

new_interval = [2, 10]

A) [[1, 12], [13, 14]]

B) [[1, 3], [2, 10], [4, 5], [7, 8], [9, 12], [13, 14]]

C) [[2, 10], [13, 14]]

Submit Answer

<


Question 1 of 4
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.

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

Iterate through the existing intervals and append all intervals occurring before the new interval to the **output** list.

Check if there is an overlap between the last interval in the **output** list and the new interval.

?

Tt

🔄

If there is an overlap, merge them by updating the end value of the last interval in the **output** list to the maximum of its current end time and the end time of the new interval.

Otherwise, append the new interval to the **output** list.

Continue iterating through the remaining intervals and merge any overlapping intervals with the last interval in the **output** list.

Return the final **output** list containing the merged intervals.



Reset

Show Solution

Submit


Try it yourself

Implement your solution in **InsertInterval.java** in the following coding playground.



InsertInterval.java
Interval.java

```
1 import java.util.*;
2 class InsertInterval {
3
4     public static List <Interval> insertInterval(List <Interval> existingInterv
5         List <Interval> output = new ArrayList <Interval> ();
6         // Write your code here
7         return output;
8     }
9 }
```



 Powered by AI  **Submit**

Test Cases Results

Case 1 Case 2 Case 3

Input #1
[[1,2],[3,4],[5,8],[9,15]]

Input #2



[2,5]

Insert Interval

← Back

Solution: Merge Interv...

Next →

Solution: Insert Interval



Mark as
Completed

