986. Interval List Intersections

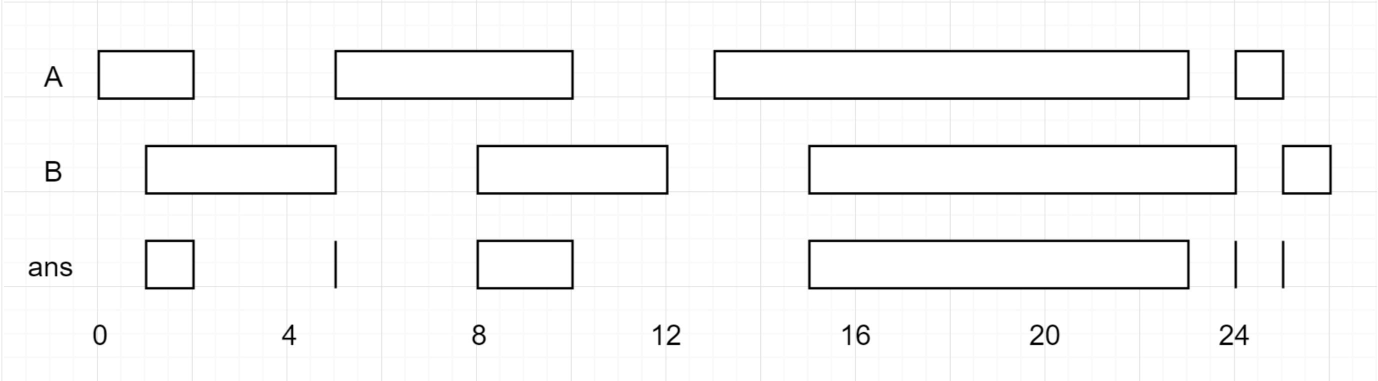
You are given two lists of closed intervals, `firstList` and `secondList`, where `firstList[i] = [start<sub>i</sub>, end<sub>i</sub>]` and `secondList[j] = [start<sub>j</sub>, end<sub>j</sub>]`. Each list of intervals is pairwise **disjoint** and in **sorted order**.

Return *the intersection of these two interval lists*.

A **closed interval** `[a, b]` (with `a <= b`) denotes the set of real numbers `x` with `a <= x <= b`.

The **intersection** of two closed intervals is a set of real numbers that are either empty or represented as a closed interval. For example, the intersection of `[1, 3]` and `[2, 4]` is `[2, 3]`.

Example 1:



```
Input: firstList = [[0,2],[5,10],[13,23],[24,25]], secondList = [[1,5],
[8,12],[15,24],[25,26]]
Output: [[1,2],[5,5],[8,10],[15,23],[24,24],[25,25]]
```

Example 2:

```
Input: firstList = [[1,3],[5,9]], secondList = []
Output: []
```

Example 3:

```
Input: firstList = [], secondList = [[4,8],[10,12]]
Output: []
```

Example 4:

```
Input: firstList = [[1,7]], secondList = [[3,10]]
Output: [[3,7]]
```

Constraints:

- `0 <= firstList.length, secondList.length <= 1000`
- `firstList.length + secondList.length >= 1`
- `0 <= start<sub>i</sub> < end<sub>i</sub> <= 10<sup>9</sup>`
- `end<sub>i</sub> < start<sub>i+1</sub>`
- `0 <= start<sub>j</sub> < end<sub>j</sub> <= 10<sup>9</sup>`
- `end<sub>j</sub> < start<sub>j+1</sub>`

```python
class Solution:
    def intervalIntersection(self, firstList: List[List[int]], secondList: List[List[int]]) -> List[List[int]]:
        ans = []



        i,j=0,0
        while i<len(firstList) and j<len(secondList):
            start = max(firstList[i][0],secondList[j][0])
            end = min(firstList[i][1],secondList[j][1])
            if start<=end:
                ans.append([start,end])

            if firstList[i][1]<secondList[j][1]:
                i = i+1
            else:
                j = j+1
        return ans




        #O(N*N)
        ans = []
        for i in range(len(firstList)):
            for j in range(len(secondList)):
                start = max(firstList[i][0],secondList[j][0])
                end = min(firstList[i][1],secondList[j][1])
                if start<=end:
                    ans.append([start,end])
        return ans
```