

{% note info %} **摘要** Title: 986. 区间列表的交集 Tag: 区间交集、双指针 Memory Limit: 64 MB Time Limit: 1000 ms {% endnote %}

Powered by:NEFU AB-IN

[Link](#)

[@TOC](#)

986. 区间列表的交集

- 题意

给定两个由一些 闭区间 组成的列表，`firstList` 和 `secondList`，其中 `firstList[i] = [starti, endi]` 而 `secondList[j] = [startj, endj]`。每个区间列表都是成对 不相交 的，并且 已经排序。返回这两个区间列表的交集。形式上，闭区间 $[a, b]$ （其中 $a \leq b$ ）表示实数 x 的集合，而 $a \leq x \leq b$ 。两个闭区间的 交集 是一组实数，要么为空集，要么为闭区间。例如， $[1, 3]$ 和 $[2, 4]$ 的交集为 $[2, 3]$ 。

- 思路

定义两个指针*i,j*分别指向A,B数组 每次取**A和B当前元素的左端点最大值**，**A和B当前元素的右端点最小值**，如果满足区间定义，那么这就是一个交区间 最后通过判断哪个元素的右端点更靠左，哪个指针右移

- 代码

```
class Solution:
    def intervalIntersection(self, firstList: List[List[int]], secondList: List[List[int]]) -> List[List[int]]:
        res = []
        i, j = 0, 0
        while i < len(firstList) and j < len(secondList):
            L, R = max(firstList[i][0], secondList[j][0]), min(firstList[i][1], secondList[j][1])
            if L <= R:
                res.append([L, R])
            if firstList[i][1] <= secondList[j][1]:
                i += 1
            else:
                j += 1
        return res
```