

0056. 合并区间

👤 [ITCharge](#) ⌚ 大约 2 分钟

- 标签：数组、排序
- 难度：中等

题目链接

- [0056. 合并区间 - 力扣](#)

题目大意

描述：给定数组 `intervals` 表示若干个区间的集合，其中单个区间为 `intervals[i] = [starti, endi]` 。

要求：合并所有重叠的区间，并返回一个不重叠的区间数组，该数组需恰好覆盖输入中的所有区间。

说明：

- $1 \leq intervals.length \leq 10^4$ 。
- $intervals[i].length == 2$ 。
- $0 \leq starti \leq endi \leq 10^4$ 。

示例：

- 示例 1:

```
输入: intervals = [[1,3],[2,6],[8,10],[15,18]]
输出: [[1,6],[8,10],[15,18]]
解释: 区间 [1,3] 和 [2,6] 重叠，将它们合并为 [1,6]。
```

py

- 示例 2:

```
输入: intervals = [[1,4],[4,5]]
输出: [[1,5]]
解释: 区间 [1,4] 和 [4,5] 可被视为重叠区间。
```

py

解题思路

思路 1：排序

1. 设定一个数组 `ans` 用于表示最终不重叠的区间数组，然后对原始区间先按照区间左端点大小从小到大进行排序。
2. 遍历所有区间。
3. 先将第一个区间加入 `ans` 数组中。
4. 然后依次考虑后边的区间：
 1. 如果第 `i` 个区间左端点在前一个区间右端点右侧，则这两个区间不会重合，直接将该区间加入 `ans` 数组中。
 2. 否则的话，这两个区间重合，判断一下两个区间的右区间值，更新前一个区间的右区间值为较大值，然后继续考虑下一个区间，以此类推。
5. 最后返回数组 `ans`。

思路 1：代码

```
class Solution:
    def merge(self, intervals: List[List[int]]) -> List[List[int]]:
        intervals.sort(key=lambda x: x[0])

        ans = []
        for interval in intervals:
            if not ans or ans[-1][1] < interval[0]:
                ans.append(interval)
            else:
                ans[-1][1] = max(ans[-1][1], interval[1])
        return ans
```

py

思路 1：复杂度分析

- 时间复杂度： $O(n \times \log_2 n)$ 。其中 n 为区间数量。
- 空间复杂度： $O(n)$ 。

