

查阅更多的题解, 请点击

Problem

56. Merge Intervals(Medium)

Given a collection of intervals, merge all overlapping intervals.

Example 1:

Input: `[[1,3],[2,6],[8,10],[15,18]]`

Output: `[[1,6],[8,10],[15,18]]`

Explanation: Since intervals `[1,3]` and `[2,6]` overlaps, merge them into `[1,6]`.

Example 2:

Input: `[[1,4],[4,5]]`

Output: `[[1,5]]`

Explanation: Intervals `[1,4]` and `[4,5]` are considered overlapping.

Solution

假设输入数组大小为N, 长度为n

$O(n\log n)$ time, $O(N)$ space

分析该题, merge interval的关键是找出overlapping intervals, 我们不可能两两比较, 很容易想到排序

note: 这里比较两intervals中的第一个元素还是第二个元素, 取决于后面遍历的方向

```
class Solution
{
public:
    vector<vector<int>> merge(vector<vector<int>> &intervals)
    {

        vector<vector<int>> res;
        int n = intervals.size();
        if (n < 1)
            return res;
        sort(intervals.begin(), intervals.end(), [](const vector<int> &a, const vector<int> &b) { return a[

        for (int i = 1; i < n; ++i)
        {
            if (intervals[i][0] <= intervals[i - 1][1])
            {
                intervals[i][0] = intervals[i - 1][0];
                intervals[i][1] = max(intervals[i][1], intervals[i - 1][1]);
            }
            else
            {
                res.push_back(intervals[i - 1]);
            }
        }
        res.push_back(intervals[n - 1]);
        return res;
    }
};
```

[GitHub传送门](#)