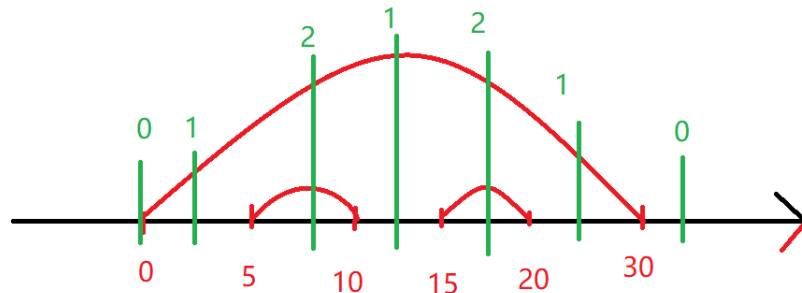


## 一、题目说明

题目253.Meeting Rooms II, 给定一系列数, 其中每个数表示会议开始、结束时间, 计算最少需要的会议室的数量。难度是Medium!

## 二、我的解答

这个题目读懂后, 在纸上画上坐标轴, 画上各个会议时间的区间, 然后从最早开始的会议开始, 重叠最大部分即是最大会议室的数量。



从“0”开始, 每次遇到会议开始时间, 则会议室的数量+1, 遇到结束, 会议室的数量-1。剩下的问题, 就是设计数据结构和算法了。

可以用map实现, 也可以用数组实现, 也可以用堆实现。

```
class Solution{
public:
    int minMeetingRooms(vector<vector<int>>& intervals){
        if(intervals.size()<1) return 0;
        int curRooms = 0;
        int result = 0;

        map<int,int> dp;
        for(auto interval: intervals){
            dp[interval[0]]++;
            dp[interval[1]]--;
        }

        for(auto iter: dp){
            curRooms = curRooms + iter.second;
            if(curRooms > result){
                result = curRooms;
            }
        }
        return result;
    }
};
```

三、优化措施

无