



Grandyang

仰天长啸仗剑红尘，冬去春来寒暑几更...

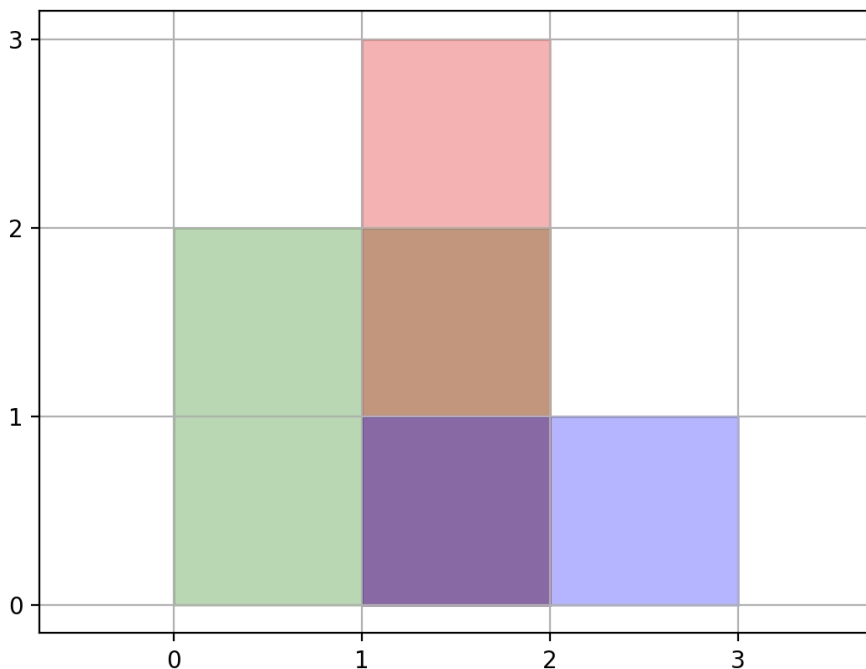
博客园 | 首页 | 新随笔 | 联系 | 订阅 | 管理

随笔 - 1444 文章 - 1 评论 - 4230

[LeetCode] 850. Rectangle Area II 矩形面积之二

We are given a list of (axis-aligned) `rectangles`. Each `rectangle[i] = [x1, y1, x2, y2]`, where (x1, y1) are the coordinates of the bottom-left corner, and (x2, y2) are the coordinates of the top-right corner of the `i`th rectangle.

Find the total area covered by all `rectangles` in the plane. Since the answer may be too large, return it modulo $10^9 + 7$.



Example 1:

```
Input: [[0,0,2,2],[1,0,2,3],[1,0,3,1]]
Output: 6
Explanation: As illustrated in the picture.
```

Example 2:

```
Input: [[0,0,1000000000,1000000000]]
Output: 49
```

公告



(请关注下方微信公众号，并留言跟博主交流)

Github同步地址，欢迎star♡

github.com/grandyang/leetcode

搜索【shua2sum】或扫描二维码
关注微信公众号【刷尽天下】

公众号

使用方法：

- 回复数字【0】随机推送一道题。
- 回复区间【1 - 919】内任意数字推送对应的题目。
- 回复关键字 例如【Two Sum】推送对应的题目。
- 回复【all】推送题目汇总列表。
- 回复【other】推送相关总结帖。
- 回复任意文字跟博主留言交流^ . ^

喜欢本博客可以请博主喝杯咖啡~



昵称：Grandyang

园龄：8年1个月

粉丝：1179

关注：36

+加关注

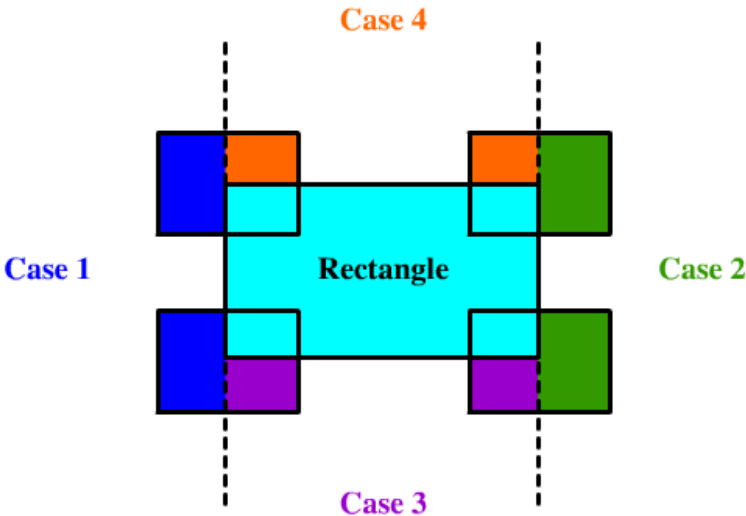
<	2020年5月						>
日	一	二	三	四	五	六	
26	27	28	29	30	1	2	
3	4	5	6	7	8	9	

Explanation: The answer is 10^18 modulo (10^9 + 7), which is (10^9)^2 = (-7)^2 = 49.

Note:

- `1 <= rectangles.length <= 200`
- `rectangles[i].length = 4`
- `0 <= rectangles[i][j] <= 10^9`
- The total area covered by all rectangles will never exceed `2^63 - 1` and thus will fit in a 64-bit signed integer.

这道题是之前那道 [Rectangle Area](http://www.cnblogs.com/grandyang/p/4563153.html) 的拓展，那道题只有两个矩形重叠，而这道题有多个矩形可能同时重叠，整体难度一下就上来了，那么通过将所有矩形面积加起来再减去重叠区域的方法这里就不太适用了，因为多个矩形在同一区域重叠的话，都减去重叠面积是会错的，还得把多减的补回来，相当的麻烦。这里我们需要换一种解题的思路，不能一股脑儿的把所有的矩形都加起来，而是应该利用微积分的思想，将重叠在一起的区域拆分成一个个的小矩形，分别累加面积，因为这里的矩形是不会旋转的，所以是可以正常拆分的。思路有了，新建一个二维数组 all 来保存所有的矩形，然后遍历给定的矩形数组，对于每个遍历到的数组，调用一个子函数，将当前的矩形加入 all 中。下面主要来看一下这个子函数 helper 该如何实现？首先要明白这个函数的作用是将当前矩形加入 all 数组中，而且用的是递归的思路，所以要传入一个 start 变量，表示当前和 all 数组中正在比较的矩形的 index，这样在开始的时候，检查一下若 start 大于等于 all 数组长度，表示已经检测完 all 中所有的矩形了，将当前矩形加入 all 数组，并返回即可。否则的话则取出 start 位置上的矩形 rec，此时就要判断当前要加入的矩形和这个 rec 矩形是否有重叠，这在 LeetCode 中有专门一道题是考察这个的 [Rectangle Overlap](https://www.cnblogs.com/grandyang/p/10367583.html)，这里用的就是那道题的判断方法，假如判断出当前矩形 cur 和矩形 rec 没有交集，就直接对 all 数组中下一个矩形调用递归函数，并返回即可。假如有重叠的话，就稍微麻烦一点，由于重叠的部位不同，所以需要分情况讨论一下，参见下图所示：



对于一个矩形 Rectangle，若有另外一个矩形跟它有重叠的话，可以将重叠区域分为四个部分，如上图的 Case1，Case2，Case3，Case4 所示，非重叠部分一定会落在一个或多个区域中，则可以把这些拆开的小矩形全部加入到矩形数组 all 中。仔细观察上图可以发现，对于将矩形 cur 拆分的情况可以分为下面四种：

- 落入区间1，条件为 $cur[0] < rec[0]$ ，产生的新矩形的两个顶点为 $\{cur[0], cur[1], rec[0], cur[3]\}$ 。
- 落入区间2，条件为 $cur[2] > rec[2]$ ，产生的新矩形的两个顶点为 $\{rec[2], cur[1], cur[2], cur[3]\}$ 。
- 落入区间3，条件为 $cur[1] < rec[1]$ ，产生的新矩形的两个顶点为 $\{max(rec[0], cur[0]), cur[1], min(rec[2], cur[2]), rec[1]\}$ 。
- 落入区间4，条件为 $cur[3] > rec[3]$ ，产生的新矩形的两个顶点为 $\{max(rec[0], cur[0]), rec[3], min(rec[2], cur[2]), cur[3]\}$ 。

这样操作下来的话，整个所有的区域都被拆分成了很多个小矩形，每个矩形之间都不会有重复，最后只要分别计算每个小矩形的面积，并累加起来就是最终的结果了，参见代码如下：

10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

搜索

最新随笔

- 1.[LeetCode] 945. Minimum Increment to Make Array Unique 使数组没有重复数字的最小增量
- 2.[AWS] Export and Import Data from DynamoDB Table 从 DynamoDB 数据库导入和导出数据
- 3.[LeetCode] 944. Delete Columns to Make Sorted 删除列使其有序
- 4.[LeetCode] 943. Find the Shortest Superstring 找到最短的超级字符串
- 5.[LeetCode] 942. DI String Match 增减DI字符串匹配
- 6.[LeetCode] 941. Valid Mountain Array 验证山形数组
- 7.[LeetCode] 940. Distinct Subsequences II 不同的子序列之二
- 8.[LeetCode] 939. Minimum Area Rectangle 面积最小的矩形
- 9.[LeetCode] 938. Range Sum of BST 二叉搜索树的区间和
- 10.[LeetCode] 937. Reorder Data in Log Files 日志文件的重新排序

积分与排名

积分 - 3037226
排名 - 17

随笔分类

- 3D Visualization(12)
- Algorithms(8)
- Amazon Web Service(4)
- C/C++, Java, Python(34)
- CareerCup(150)
- CUDA/OpenCL(1)
- Digital Image Processing(3)
- Entertainment(6)
- GTK+/VTK/ITK/FLTK(20)
- IOS(7)
- LaTeX(3)
- LeetCode(916)
- LintCode(101)
- MatLab(10)
- Maya / 3ds Max(10)
- MySQL(2)
- Node.js / JavaScript(8)
- OpenCV(37)
- Point Grey Research(11)
- Qt(49)
- Software/Tools(3)
- Useful Links(34)

解法一：

```
class Solution {
public:
    int rectangleArea(vector<vector<int>>& rectangles) {
        long res = 0, M = 1e9 + 7;
        vector<vector<int>> all;
        for (auto rectangle : rectangles) {
            helper(all, rectangle, 0);
        }
        for (auto &a : all) {
            res = (res + (long)(a[2] - a[0]) * (long)(a[3] - a[1])) % M;
        }
        return res;
    }
    void helper(vector<vector<int>>& all, vector<int> cur, int start) {
        if (start >= all.size()) {
            all.push_back(cur); return;
        }
        auto rec = all[start];
        if (cur[2] <= rec[0] || cur[3] <= rec[1] || cur[0] >= rec[2] || cur[1] >= rec[3]) {
            helper(all, cur, start + 1); return;
        }
        if (cur[0] < rec[0]) {
            helper(all, {cur[0], cur[1], rec[0], cur[3]}, start + 1);
        }
        if (cur[2] > rec[2]) {
            helper(all, {rec[2], cur[1], cur[2], cur[3]}, start + 1);
        }
        if (cur[1] < rec[1]) {
            helper(all, {max(rec[0], cur[0]), cur[1], min(rec[2], cur[2]), rec[1]}, start + 1);
        }
        if (cur[3] > rec[3]) {
            helper(all, {max(rec[0], cur[0]), rec[3], min(rec[2], cur[2]), cur[3]}, start + 1);
        }
    }
};
```

下面这种解法更是利用了微积分的原理，把x轴长度为1当作一个步长，然后计算每一列有多少个连续的区间，每个连续区间又有多少个小正方形，题目中给的例子每一个列都只有一个连续区间，但事实上是可以有多个的，只要算出了每一列 1x1 小正方形的个数，将所有列都累加起来，就是整个区域的面积。这里求每列上小正方形个数的方法非常的 tricky，博主也不知道该怎么讲解，大致就是要求同一列上每个连续区间中的小正方形个数，再累加起来。对于每个矩形起始的横坐标，映射较低的y值到1，较高的y值到-1，对于结束位置的横坐标，刚好反过来一下，映射较低的y值到-1，较高的y值到1。这种机制跟之前那道 [The Skyline Problem] (<http://www.cnblogs.com/grandyang/p/4534586.html>) 有些异曲同工之妙，都还是为了计算高度差服务的。要搞清楚这道题的核心思想，不是一件容易的事，博主的建议是就拿题目中给的例子带入到下面的代码中，一步一步执行，并分析结果，是能够初步的了解解题思路的，若实在有理解上的问题，博主可以进一步写些讲解，参见代码如下：

解法二：

```
class Solution {
public:
    int rectangleArea(vector<vector<int>>& rectangles) {
        long res = 0, pre_x = 0, height = 0, start = 0, cnt = 0, M = 1e9 + 7;
        map<int, vector<pair<int, int>>> groupMap;
        map<int, int> cntMap;
        for (auto &a : rectangles) {
            groupMap[a[0]].push_back({a[1], 1});
            groupMap[a[0]].push_back({a[3], -1});
            groupMap[a[2]].push_back({a[1], -1});
            groupMap[a[2]].push_back({a[3], 1});
        }
        for (auto &group : groupMap) {
```

随笔档案

2020年5月(4)
2020年4月(5)
2020年3月(3)
2020年2月(5)
2020年1月(3)
2019年12月(2)
2019年11月(4)
2019年10月(9)
2019年9月(8)
2019年8月(10)
2019年7月(8)
2019年6月(13)
2019年5月(16)
2019年4月(14)
2019年3月(10)
2019年2月(12)
2019年1月(10)
2018年12月(8)
2018年11月(19)
2018年10月(9)
2018年9月(6)
2018年8月(8)
2018年7月(11)
2018年6月(10)
2018年5月(11)
2018年4月(13)
2018年3月(15)
2018年2月(14)
2018年1月(17)
2017年12月(12)
2017年11月(16)
2017年10月(29)
2017年9月(21)
2017年8月(10)
2017年7月(12)
2017年6月(21)
2017年5月(26)
2017年4月(18)
2017年3月(22)
2017年2月(23)
2017年1月(13)
2016年12月(26)
2016年11月(30)
2016年10月(30)
2016年9月(24)
2016年8月(40)
2016年7月(31)
2016年6月(33)
2016年5月(30)
2016年4月(70)
2016年3月(32)
2016年2月(32)
2016年1月(25)
2015年12月(3)
2015年11月(36)
2015年10月(43)
2015年9月(51)
2015年8月(46)
2015年7月(45)
2015年6月(29)
2015年5月(28)

赞助

```
res = (res + (group.first - pre_x) * height) % M;
for (auto &a : group.second) {
    cntMap[a.first] += a.second;
}
height = 0, start = 0, cnt = 0;
for (auto &a : cntMap) {
    if (cnt == 0) start = a.first;
    cnt += a.second;
    if (cnt == 0) height += a.first - start;
}
pre_x = group.first;
}
return res;
};
```

Github 同步地址:

<https://github.com/grandyang/leetcode/issues/850>

类似题目：

[Rectangle Overlap](#)

[Rectangle Area](#)

[The Skyline Problem](#)

参考资料：

<https://leetcode.com/problems/rectangle-area-ii/>

<https://leetcode.com/problems/rectangle-area-ii/discuss/138028/Clean-Recursive-Solution-Java>

<https://leetcode.com/problems/rectangle-area-ii/discuss/214365/Short-C%2B%2B-solution.-EZ-to-understand.-Beat-99>

[LeetCode All in One 题目讲解汇总(持续更新中...)](<https://www.cnblogs.com/grandyang/p/4606334.html>)

分类: [LeetCode](#)

好文要顶

关注我

收藏该文

Grandyang

关注 - 36

粉丝 - 1179

+加关注

« 上一篇：[\[LeetCode\] 910. Smallest Range II 最小区间之二](#)

» 下一篇：[\[LeetCode\] 1123. Lowest Common Ancestor of Deepest Leaves 最深叶节点的最小公共父节点](#)
posted @ 2019-08-18 02:56 Grandyang 阅读(1289) 评论(0) 编辑 收藏

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)， [访问](#) [网站首页](#)。

- 【推荐】了解你才能更懂你，博客园首发问卷调查，助力社区新升级
- 【推荐】超50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库
- 【推荐】合辑 | 学习python不可不知的开发者词条汇总！
- 【推荐】阿里技术3年大合辑免费电子书一键下载

- 2015年4月(42)
- 2015年3月(55)
- 2015年2月(61)
- 2015年1月(27)
- 2014年12月(8)
- 2014年11月(27)
- 2014年10月(35)
- 2014年9月(5)

最新评论

1. Re:[LeetCode] 6. ZigZag Conve
rsion 之字型转换字符串
@c2156959 我帮博主回一下吧。
因为在 for loop 里面，两个 for 循
环一共会读 2*numRows - 2 个数，
而 i < n 是防止数组越界，比如：
字符串 s.size() =...

--星海寻梦233

2. Re:[LeetCode] Delete Node
Linked List 删除链表的节点
被这道题骚到了

--b...

3. Re:[LeetCode] 169. Majority Ele
ment 求大多数
@逆光也很美 对的，必须保证过半
才行（正好等于一半也是可以
的）。只满足众数是绝对不行的。
给两个例子：例1：vector<int> n
ums = { 2,2,2,3,3,5}，返回为2，正
确。例2...

--zxc120389574

4. Re:LeetCode All in One 题目讲
解汇总(持续更新中...)
厉害了，像楼主学习

--DayDayUp~~

5. Re:[LeetCode] 138. Copy List w
ith Random Pointer 拷贝带有随机
指针的链表
个人感觉解法一的这里可能有个小
bug：node = res; cur = head; wh
ile (cur) { node->random = m[cur->
random]; node = no...

--bw98

阅读排行榜

1. LeetCode All in One 题目讲解汇
总(持续更新中...)(850161)
2. [LeetCode] 1. Two Sum 两数之
和(121524)
3. [LeetCode] 15. 3Sum 三数之和
(71837)
4. Manacher's Algorithm 马拉车算
法(66921)
5. [LeetCode] 4. Median of Two So
rted Arrays 两个有序数组的中位数
(57287)
6. [LeetCode] 5. Longest Palindro
mic Substring 最长回文子串(5702
3)