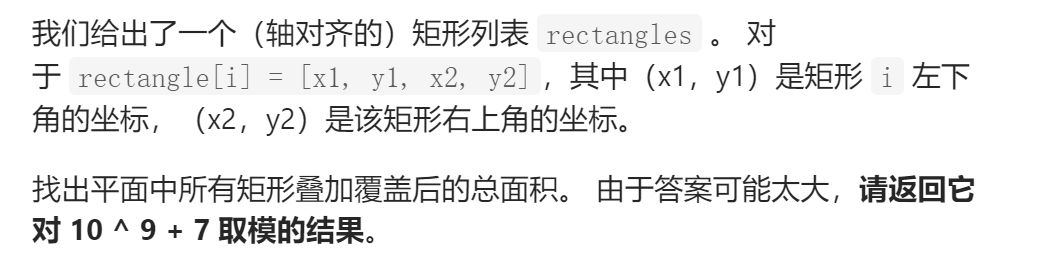
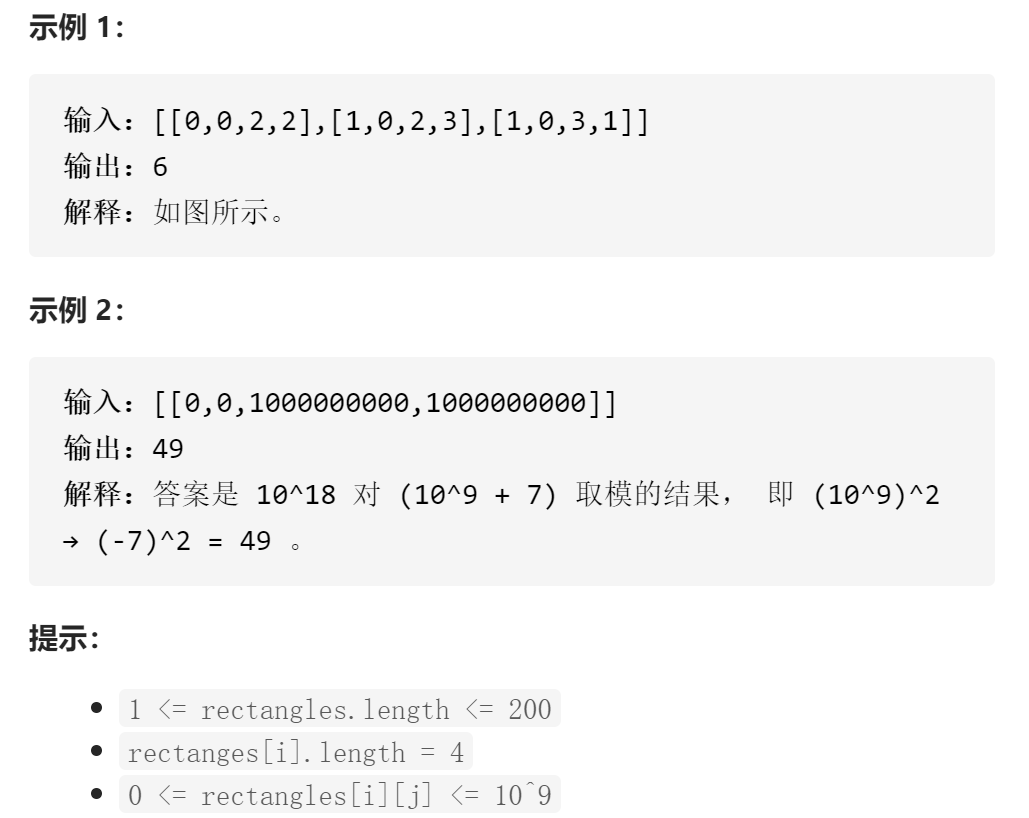
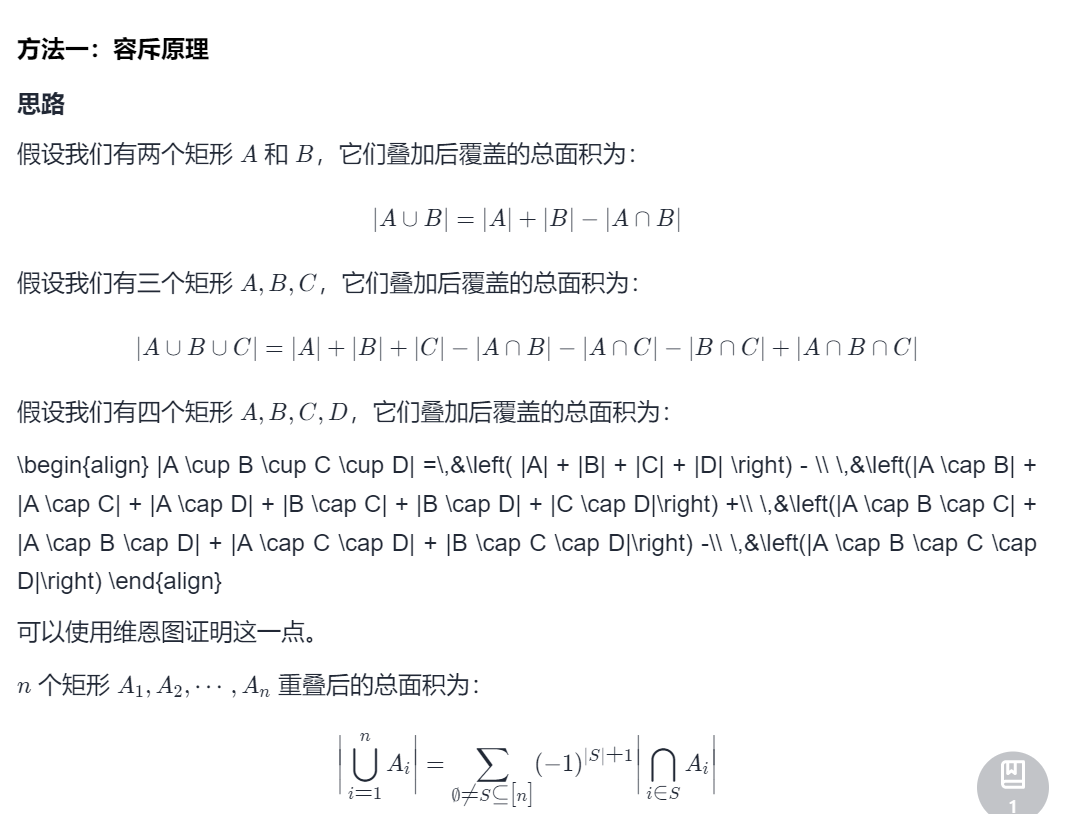
# [850. 矩形面积 II](https://leetcode-cn.com/problems/rectangle-area-ii/)

## 1题目描述





## 我的思路：回溯算法，暴力求解【有关于容斥原理的排列组合】



class Solution {  
private:   
typedef long long int ll;  
long long mod=1000000007;  
long long int sum;//结果  
vector<bool>tags;//标志数组  
  
void backtracking(int start,vector<vector<int>>& rectangles,bool flag,vector<ll>datas){  
 //当flag为1时表示加，为0时表示减  
 //终止条件-无  
 for(int i=start;i<rectangles.size();i++){  
 if(tags[i])continue;  
 ll x1=max(datas[0],(ll)rectangles[i][0]);  
 ll y1=max(datas[1],(ll)rectangles[i][1]);  
 ll x2=min(datas[2],(ll)rectangles[i][2]);  
 ll y2=min(datas[3],(ll)rectangles[i][3]);  
 if(x1>=x2||y1>=y2)continue;  
 ll temp=(x2-x1)\*(y2-y1);  
 if(flag)sum+=temp;  
 else sum-=temp;  
 sum%=mod;  
 vector<ll>temps={x1,y1,x2,y2};  
 tags[i]=true;  
 backtracking(i+1,rectangles,!flag,temps);  
 tags[i]=false;  
 }  
}  
  
public:  
 int rectangleArea(vector<vector<int>>& rectangles) {  
 //容斥原理，排列组合  
 //初始化  
 sum=0;  
 tags.resize(rectangles.size(),false);  
 //此处求出第一个的面积后往下递归  
 int i=0;  
 for(auto temp:rectangles){  
 tags[i]=true;  
 ll x1=temp[0];  
 ll y1=temp[1];  
 ll x2=temp[2];  
 ll y2=temp[3];  
 sum+=(x2-x1)\*(y2-y1);  
 sum%=mod;  
 vector<ll>temps={x1,y1,x2,y2};  
 backtracking(i+1,rectangles,false,temps);  
 tags[i]=false;  
 ++i;  
 }  
 return sum;  
 }  
};

## attention:10^9+7中^在语言中表示异或。应该写成：1000000007；或写成 int MOD = 1e9 + 7;