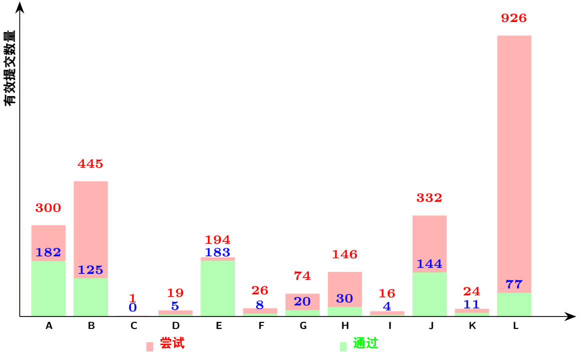
2024 国际大学生程序设计竞赛（ICPC）湖北省省赛试题讲解

华中科技大学 ACM-ICPC 集训队
华中科技大学网络空间安全学院 ACMS 团队

2024 年 4 月 27 日

赛时数据

- 截至封榜前 4 小时，总共接收 183 队共 2503 次有效提交，其中 789 次正确提交。
- 各题提交情况如下：



- 将按照试题通过数量顺序进行讲题。

E. Spicy or Grilled?

题目大意

n 个人，有 x 个人想吃板烧鸡腿堡，剩下的人吃麦辣鸡腿堡。已知麦辣鸡腿堡每套 a 元，板烧鸡腿堡每套 b 元，计算总花费

$1 \leq n, a, b \leq 10^4, 0 \leq x \leq n$

E. Spicy or Grilled?

题目大意

n 个人，有 x 个人想吃板烧鸡腿堡，剩下的人吃麦辣鸡腿堡。已知麦辣鸡腿堡每套 a 元，板烧鸡腿堡每套 b 元，计算总花费
$1 \leq n, a, b \leq 10^4, 0 \leq x \leq n$

签到题。

输出 $(n - x) \times a + x \times b$。

A. Long Live

题目大意

给两个正整数 x, y，找到两个正整数 a, b，使得

$$\sqrt{\frac{\operatorname{lcm}(x, y)}{\gcd(x, y)}} = a\sqrt{b}, \tag{1}$$

并且最大化 $a \cdot b$ 的值。
$1 \leq x, y \leq 10^9$

A. Long Live

题目大意

给两个正整数 x, y，找到两个正整数 a, b，使得

$$\sqrt{\frac{\operatorname{lcm}(x, y)}{\gcd(x, y)}} = a\sqrt{b}, \tag{1}$$

并且最大化 $a \cdot b$ 的值。
$1 \le x, y \le 10^9$

对于式子 (1) 而言，等式左边可以直接算出。

将 $a \cdot b$ 变形成为 $\frac{(a\sqrt{b})^2}{a}$，将上式代入，得 $a \cdot b = \frac{\operatorname{lcm}(x,y)}{a \cdot \gcd(x,y)}$。故 $a = 1$ 时，$a \cdot b$ 可以有最大值，此时 $b = \frac{\operatorname{lcm}(x,y)}{\gcd(x,y)}$。

J. Points on the Number Axis A

题目大意

n 个数，每次随机选择两个数 a,b，将他们删除并加入 $\frac{a+b}{2}$，求最后剩下的数的期望。
$1 \le n \le 10^6$

J. Points on the Number Axis A

题目大意

n 个数，每次随机选择两个数 a, b，将他们删除并加入 $\frac{a+b}{2}$，求最后剩下的数的期望。
$1 \leq n \leq 10^6$

~~省流：答案是平均数。~~

可以看出每次选择时，每个数被选到的概率是一致的（$\frac{2}{n}$），推测答案是平均数~~，然后就过了~~。

数学归纳法：
计 s 为所有数之和。
$n = 2$ 时答案是平均数。
若 $n = k - 1$ 时答案是平均数，则 $n = k$ 时，有:

$$ans = \frac{2}{k(k-1)} \left(\frac{1}{k-1} \sum_{i=1}^{k} \sum_{j=i+1}^{k} \left(s - \frac{a_i + a_j}{2} \right) \right)$$
$$= \frac{2}{k(k-1)^2} \left(\frac{k(k-1)}{2} s - (k-1) \frac{s}{2} \right) = \frac{s}{k}.$$

证毕。

B. Nana Likes Polygons

题目大意

给出 2D 平面上的 n 个点，选出若干个点作为顶点构成一个凸多边形。求所有合法的凸多边形的最小面积。
$1 \leq n \leq 100$

B. Nana Likes Polygons

题目大意

给出 2D 平面上的 n 个点，选出若干个点作为顶点构成一个凸多边形。求所有合法的凸多边形的最小面积。
$1 \leq n \leq 100$

可以发现面积最小的凸多边形一定是三角形。

枚举三角形的三个顶点 A, B 和 C，若 ΔABC 不退化，那么用叉积计算面积 $\frac{1}{2}|\overrightarrow{AB} \times \overrightarrow{AC}|$。找到最小的值即可。

L. LCMs

题目大意

有一张 $n-1$ 个点的完全图，节点编号是 $2 \sim n$，u 和 v 之间的边权是 $\operatorname{lcm}(u,v)$。T 次询问从 x 走到 y 的最小代价。
$1 \le n \le 10^7, 1 \le T \le 1000$

L. LCMs

题目大意

有一张 $n-1$ 个点的完全图，节点编号是 $2\sim n$，u 和 v 之间的边权是 $\mathrm{lcm}(u,v)$。T 次询问从 x 走到 y 的最小代价。
$1\le n\le 10^7, 1\le T\le 1000$

首先 $x=y$，输出 0。

不妨假设 $x<y$。

- 当 y 是 x 的倍数的情况下，费用是 y；
- 当 y 和 x 不互质的情况下，可以 $x\to\gcd(x,y)\to y$，费用是 $x+y$；
- 当 x 和 y 互质的情况下：如果 x 走了若干步走到了 z，然后 z 走到了 y，根据 lcm 的性质，z 一定是一个质数。
同样的，中间经过的点一定都是质数。
因为 x,y 互质，所以他们的质因子均不相同，所以经过的质数越小越好，比如 x,y 的最小质因数和 2。

所以本题只需要 5 个数字使用最短路算法计算，或者采用分类讨论均可。

H. Genshin Impact Startup Forbidden III

题目大意

$n \times m$ **池塘**，k **块池塘有鱼**，(x_i, y_i) **有** a_i **条鱼**。
用十字形炸弹炸鱼，最少需要的炸弹数目。
$1 \leq n, m \leq 1000, 1 \leq k \leq 10, 1 \leq a_i \leq 3$

H. Genshin Impact Startup Forbidden III

题目大意

$n \times m$ 池塘，k 块池塘有鱼，(x_i, y_i) 有 a_i 条鱼。
用十字形炸弹炸鱼，最少需要的炸弹数目。
$1 \leq n, m \leq 1000, 1 \leq k \leq 10, 1 \leq a_i \leq 3$

对于一个有鱼的池塘，有周围与自己本身五个关键位置可以捕获当前位位置的鱼。

把这些位置存储到 `unordered_set` 中。

用四进制数 S 表示每块池塘中剩余的鱼的数目，$dp[S]$ 表示达成该状态最少的炸弹数。

枚举所有的关键位置，计算状态的转移。

G. Genshin Impact Startup Forbidden II

题目大意

围棋 m 步行棋，给出每一步行棋后，黑子/白子被提数目。
$1 \leq m \leq 5 \times 10^5$

G. Genshin Impact Startup Forbidden II

题目大意

围棋 m 步行棋，给出每一步行棋后，黑子/白子被提数目。
$1 \leq m \leq 5 \times 10^5$

bfs 计算每个连通块的气，若为 0，则删去。

时间复杂度 $O(s^2 m)$。

思考：如果只允许 $O(m \cdot \alpha(m))$？

并查集维护连通性与气，暴力删棋子。

K. Points on the Number Axis B

题目大意

n 个有顺序的数，每次随机选择两个相邻的数 a, b，将他们删除并加入 $\frac{a+b}{2}$，求最后剩下的数的期望。

$1 \leq n \leq 10^6$

K. Points on the Number Axis B

题目大意

n 个有顺序的数，每次随机选择两个相邻的数 a, b，将他们删除并加入 $\frac{a+b}{2}$，求最后剩下的数的期望。
$1 \leq n \leq 10^6$

不妨考虑每个数的贡献。

计 $f_{i,j}$ 表示左边有 i 个数，右边有 j 个数时的贡献系数。有：

$$\begin{aligned} f_{i,j} &= \frac{i-1}{i+j} \cdot f_{i-1,j} + \frac{1}{2} \cdot \frac{1}{i+j} \cdot f_{i-1,j} + \frac{j-1}{i+j} \cdot f_{i,j-1} + \frac{1}{2} \cdot \frac{1}{i+j} \cdot f_{i,j-1} \\ &= \frac{1}{i+j} \cdot \left(\left(i - \frac{1}{2} \right) f_{i-1,j} + \left(j - \frac{1}{2} \right) f_{i,j-1} \right) \end{aligned}$$

$$(i+j)! \cdot f_{i,j} = (i+j-1)! \cdot \left(\left(i - \frac{1}{2} \right) f_{i-1,j} + \left(j - \frac{1}{2} \right) f_{i,j-1} \right)$$

令 $g_{i,j} = (i+j)! f_{i,j}$，有：$g_{i,j} = (i - \frac{1}{2}) g_{i-1,j} + (j - \frac{1}{2}) g_{i,j-1}$

把 $g_{i,j}$ 看成网格图，那么问题实质上就转换为网格图路径计数。并且我们发现，路径的数量与路径的形状无关，因而可以使用组合数处理。预处理 $\prod(i - \frac{1}{2})$ 与 $i!$，时空复杂度 $O(n)$，可以通过。

F. Enchanted

题目大意

给一个值域为 $[1,30]$，长度为 n 的序列，然后定义一个合并操作为用两个相同的元素（不妨设为 l）合并成一个新的元素，新元素的值为 $l+1$，同时代价为 2^{l+1}。然后需要维护 m 个操作：

- **操作 1** 求出区间 $[l,r]$ 内的元素通过任意次合并操作能得到的最大元素。
- **操作 2** 将区间 $[l,r]$ 内的元素合并，直到每个元素都互不相同，然后再加入一个值为 k 的元素，并再继续合并直到不存在相同元素，求出加入新元素之后所有合并的代价之和。
- **操作 3** 将第 pos 个元素的值改成 k。
- **操作 4** 回到第 t 次操作结束后的状态。

$1 \le n \le 10^6, 1 \le m \le 10^6$

F. Enchanted

题目大意

给一个值域为 $[1,30]$，长度为 n 的序列，然后定义一个合并操作为用两个相同的元素（不妨设为 l）合并成一个新的元素，新元素的值为 $l+1$，同时代价为 2^{l+1}。然后需要维护 m 个操作：

- 操作 1 求出区间 $[l,r]$ 内的元素通过任意次合并操作能得到的最大元素。
- 操作 2 将区间 $[l,r]$ 内的元素合并，直到每个元素都互不相同，然后再加入一个值为 k 的元素，并再继续合并直到不存在相同元素，求出加入新元素之后所有合并的代价之和。
- 操作 3 将第 pos 个元素的值改成 k。
- 操作 4 回到第 t 次操作结束后的状态。

$1 \le n \le 10^6, 1 \le m \le 10^6$

做法 1 由于书的惩罚只会出现不到 60 次，并且具有可加性。所以考虑维护值域在 1 到 2^{60} 大小的主席树。对于操作 1，转换成求区间和操作；对于操作 2，转换成求区间和通过位运算计算具体数值；对于操作 3，单点修改；操作 4 直接回到历史版本。令 $N=60$（维护值域上限的对数）。时间复杂度 $O(m(N+\log n))$；

F. Enchanted

题目大意

给一个值域为 $[1,30]$，长度为 n 的序列，然后定义一个合并操作为用两个相同的元素（不妨设为 l）合并成一个新的元素，新元素的值为 $l+1$，同时代价为 2^{l+1}。然后需要维护 m 个操作：

- **操作 1** 求出区间 $[l,r]$ 内的元素通过任意次合并操作能得到的最大元素。
- **操作 2** 将区间 $[l,r]$ 内的元素合并，直到每个元素都互不相同，然后再加入一个值为 k 的元素，并再继续合并直到不存在相同元素，求出加入新元素之后所有合并的代价之和。
- **操作 3** 将第 pos 个元素的值改成 k。
- **操作 4** 回到第 t 次操作结束后的状态。

$1 \le n \le 10^6, 1 \le m \le 10^6$

- **做法 1** 由于书的惩罚只会出现不到 60 次，并且具有可加性。所以考虑维护值域在 1 到 2^{60} 大小的主席树。对于操作 1，转换成求区间和操作；对于操作 2，转换成求区间和通过位运算计算具体数值；对于操作 3，单点修改；操作 4 直接回到历史版本。令 $N=60$（维护值域上限的对数）。时间复杂度 $O(m(N+\log n))$；
- **做法 2** 考虑离线维护，由于只有操作 4 比较难以维护，可以通过 dfs 序来维护历史版本。从而可以维护 N 个树状数组来达成目标。朴素做法是 $O(N^2 \times m)$，会被卡。我们可以考虑和做法 1 相同的值域维护降低复杂度到 $O(m(N+\log n))$。

I. Colorful Tree

题目大意

给出一棵 n 个顶点的树，每个点初始为白色。共 q 次操作，每次操作将一条链染黑并求最长同色链（操作不独立）。

$1 \leq n, q \leq 2 \times 10^5$

I. Colorful Tree

题目大意

给出一棵 n 个顶点的树，每个点初始为白色。共 q 次操作，每次操作将一条链染黑并求最长同色链（操作不独立）。
$1 \le n, q \le 2 \times 10^5$

实际上就是求分别求黑点和白点的直径。

染色操作是永久的，可以维护每个点变黑的时间（同理也就是变白的时间），就变为按时间顺序加点维护直径，并查集维护，求白点直径时，倒着做一遍即可。

时间复杂度 $O(n \log n)$。

D. MACARON Likes Happy Endings

题目大意

给定一个长度为 n 的序列 $\{a_i\}$，现在将其划分为连续的 k 段。定义每一段的贡献是这一段内所有异或和为 d 的连续子段个数。最小化所有 k 段的贡献和。

$1 \le n \le 10^5, 1 \le k \le \min(n, 20), 0 \le a_i, d \le 10^6$

D. MACARON Likes Happy Endings

题目大意

给定一个长度为 n 的序列 $\{a_i\}$，现在将其划分为连续的 k 段。定义每一段的贡献是这一段内所有异或和为 d 的连续子段个数。最小化所有 k 段的贡献和。
$1 \le n \le 10^5, 1 \le k \le \min(n, 20), 0 \le a_i, d \le 10^6$

考虑 DP。记 $f_{i,j}$ 表示前 i 个数划分为了 j 段得到的最小答案。

可以得到 $O(n^2k)$ 的转移

$$f_{i,j} = \min_{1 \le p \le i} \{f_{p-1,j-1} + w(p, i)\}, \tag{2}$$

其中 $w(l, r)$ 表示原序列区间 $[l, r]$ 这一段的异或和等于 d 的子段个数。

计算 w 可以移动端点开桶来记录贡献，这样就可以做到边枚举转移点边维护 w 值。具体地，令 sum_i 为数组 a_i 的前缀和。假设现在维护的是 $w(l, r)$ 的值，开一个桶 cnt，表示前缀和的值在区间内对应的个数，即维护 $\mathrm{sum}_i, i \in [l-1, r]$。

考虑当 $[l, r]$ 扩充为 $[l, r+1]$ 时，那么会产生 $r+1-l+1$ 个子区间，现在需要统计有多少个 $i \in [l-1, r]$ 使得 $\mathrm{sum}_i \oplus \mathrm{sum}_{r+1} = d \Rightarrow \mathrm{sum}_i = \mathrm{sum}_{r+1} \oplus d$。因此查询桶内 $\mathrm{sum}_{r+1} \oplus d$ 值的个数，作为增量得到 $w(l, r+1)$ 的值。
随后，将 sum_{r+1} 添加至桶内。
故在枚举 DP 状态时，可以 $O(1)$ 维护 w。

D. MACARON Likes Happy Endings

考虑优化，我们发现对于 w 这一函数，可以证明 $\forall a \le b \le c \le d, w(a,d) + w(b,c) \ge w(a,c) + w(b,d)$。
这是因为不等号左边的式子计算时一定比右侧多考虑 $(d-c)(b-a)$ 个区间，因此异或和 $= d$ 的区间个数不会少于右边。

接下来的引理 1，能帮助我们优化枚举决策点。

引理 1 (决策单调性)

若 w 满足不等式 $\forall a \le b \le c \le d, w(a,d) + w(b,c) \ge w(a,c) + w(b,d)$，并且 *DP* 方程为式 (2)，*DP* 状态 $f_{i,j}$ 中 i 具有决策单调性。
即固定 j 时，若 $f_{i,j}$ 的最优决策点为 p_i，那么 $p_{i+1} \ge p_i$。

$$f_{i,j} = \min_{1 \le p \le i} \{f_{p-1,j-1} + w(p,i)\} \tag{2}$$

D. MACARON Likes Happy Endings

简记 $f_{i,j-1}$ 为 f_i。

证明.

设 p_i 表示 i 的最优决策点是 p_i。那么有

$$f_{p_i} + w(p_i, i) \le f_j + w(j, i) \tag{3}$$

同时，我们需要证明的式子是

$$f_{p_{i+1}} + w(p_{i+1}, i+1) \le f_j + w(j, i+1) \tag{4}$$

将式 (4) 中 j 变为 p_i

$$f_{p_{i+1}} + w(p_{i+1}, i+1) \le f_{p_i} + w(p_i, i+1) \tag{5}$$

下面需要分情况讨论

1. $p_{i+1} = i$，那么显然 $p_{i+1} = i \ge p_i$ 的，满足决策单调性；

D. MACARON Likes Happy Endings

证明 (续).

$$f_{p_i} + w(p_i, i) \le f_j + w(j, i) \tag{3}$$

$$f_{p_{i+1}} + w(p_{i+1}, i+1) \le f_{p_i} + w(p_i, i+1) \tag{5}$$

下面需要分情况讨论

1. $p_{i+1} = i$，那么显然 $p_{i+1} = i \ge p_i$ 的，满足决策单调性；
2. $p_{i+1} \ne i$，将式 (3) 中 j 变为 p_{i+1}，那么

   $$f_{p_i} + w(p_i, i) \le f_{p_{i+1}} + w(p_{i+1}, i) \tag{6}$$

   式 (5)+ 式 (6) 得

   $$w(p_{i+1}, i+1) + w(p_i, i) \le w(p_i, i+1) + w(p_{i+1}, i) \tag{7}$$

   若 $p_i > p_{i+1}$，代入式 (7)，可以发现不等号与原四边形不等式不等号方向相反，显然矛盾。因此 $p_i \le p_{i+1} \le i \le i+1$。

综上满足 $p_i \le p_{i+1}$，即满足决策单调性。

D. MACARON Likes Happy Endings

假设我们已经有了划分为 $j-1$ 段的 DP 值 g_i，即 $g_i \stackrel{\text{def}}{=} f_{i,j-1}$。现在考虑如何用 g 的值计算划分为 j 段时的 DP 值 $f_i \stackrel{\text{def}}{=} f_{i,j}$。

此时 DP 方程满足

$$f_i = \min_{1 \le p \le i} \{g_{p-1} + w(p, i)\}$$

考虑使用分治的方法来解决。
例如，考虑我们需要得到 $[L, R]$ 这个区间的 DP 值，并且备选的决策区间是 $[l, r]$。若对于 $MID\left(MID = \frac{L+R}{2}\right)$ 这个位置的 DP 值取最优时的决策点是 loc，那么由决策单调性，$[L, MID-1]$ 的备选区间就是 $[l, \mathrm{loc}]$，而 $[MID+1, R]$ 的备选区间是 $[\mathrm{loc}, r]$。这样把问题分成两个子问题分治递归下去。

而如何找到 loc，方法是直接暴力扫 $[l, r]$ 中所有点作为决策点去转移到 MID 上，找到其中的最优值。以 CDQ 分治的思想，整个算法时间复杂度是 $O(n \log n)$ 的。

由于转移时的 f 取值是不会受其他位置的 f 值的影响的，这样就保证了每个 f 的计算不需要先决条件，这样就能保证分治的正确性。

最后我们只需要在最外层枚举划分的段数 j，每次将上一轮得到的 DP 值作为这一轮的 g 数组进行 CDQ 分治即可计算这一轮的答案。总复杂度 $O(nk \log n)$。

C. Lili Likes Polygons

题目大意

给定平面上的 k 个矩形，记这些矩形的并为 P。现在需要找出最少的不相交的矩形，使得这些矩形能够恰好覆盖 P。输出这个最小值。

$1 \leq k \leq 300$，保证 P 的顶点个数不超过 2000

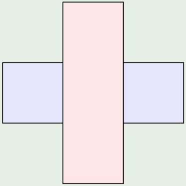
C. Lili Likes Polygons

题目大意

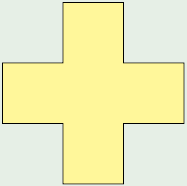
给定平面上的 k 个矩形，记这些矩形的并为 P。现在需要找出最少的不相交的矩形，使得这些矩形能够恰好覆盖 P。输出这个最小值。
$1 \leq k \leq 300$，保证 P 的顶点个数不超过 2000

示例

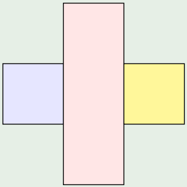
输入:



区域 P:



输出: 3
一个合法的解:



C. Lili Likes Polygons

首先假设 P 是连通的，即 P 是一个带孔的多边形。定义**多边形的大小**为它的顶点个数 $|P|$。
若顶点 p 对应的内角和为 $90°$，我们称其为**凸点**；否则称其为**凹点**（$270°$），并记凹点构成的集合为 C。

如果我们将在多边形内部分割开矩形的线称为"**割线**"，接下来考虑极小的矩形划分具有的一些性质。

引理 2

对于任意一个 P 的极小的矩形划分，所有的割线至少有一个顶点是 P 上的凹点，即存在一个端点 $\in C$。

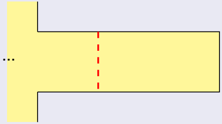
C. Lili Likes Polygons

引理 2

对于任意一个 P 的极小的矩形划分，所有的割线至少有一个顶点是 P 上的凹点，即存在一个端点 $\in C$。

证明.

如果存在一条割线的两个端点都不是凹点的话，删除这条线能够合并相邻的两个矩形。这样能够得到一个更小的合法解，与极小矛盾。



C. Lili Likes Polygons

我们令 $\mathcal{R}$ 表示一个合法的矩形剖分。
对于 $\mathcal{R}$ 中的割线，若割线连接了两个凹点，我们称其为“**好割线**”。

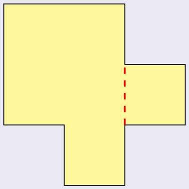
令 N 表示最大的不相交的**好**割线集合。

引理 3

对于任意一个 P 的极小矩形剖分 $\mathcal{R}$，以及关于 $\mathcal{R}$ 的最大的不相交的**好**割线集合 N，那么有

$$4|\mathcal{R}| = |P| + 2|C| - 4|N|. \tag{8}$$

证明.



考虑 P 的内角和，它等于 $90^\circ\times$ 凸点数 $+270^\circ\times$ 凹点数 $= (|P| + 2|C|) \times 90^\circ$。接下来，我们计算切割矩形过程中对内角和的增量。
对于 N 中的每一条割线 $e \in N$，e 连接两个凹点，每个端点被切成两部分：90° 和 180°。前者形成 $\mathcal{R}$ 中最终矩形的内角，而后者构成了一条边。因此，e 产生的增量是 $-2 \times 180^\circ = -360^\circ$。因为 N 中的边是不相交的，所以每条边总是可以切割两个凹角，并获得 -360° 的增量。因此，N 的增量是 $-4|N| \times 90^\circ$。

C. Lili Likes Polygons

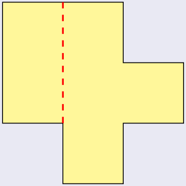
证明 (续).

对于不在 N 中的其余割线，由于 $\mathcal{R}$ 是极小的，根据引理 2，这些割线至少有一个凹端点。对于 C 中的每条边 $e=(c,p), c \in C$ 有两种情况：

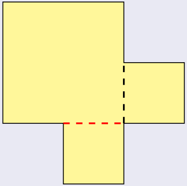
1. $p \notin C$. 在这种情况下，p 位于 P 的一条边或割线上，它将被分成两个凸点，因此内角和增加了 $180°$ 。c 也切割成两部分：$90°$ 和 $180°$，所以增量等于 $180° - 180° = 0$。
2. $p \in C$. 由于 N 是最大化的，且 $e \notin N$，因此 p 已被其他割线分割。此时，增量与第一种情况相同，等于 0。

因此，$\mathcal{R}$ 中矩形的内角和只取决于 N 的大小。由于内角和等于 $4|\mathcal{R}| \times 90°$，所以等式 $4|\mathcal{R}| \times 90° = (|P| + 2|C|) \times 90° - 4|N| \times 90°$ 成立，因此式 (8) 成立。

情况 1：



情况 2：



C. Lili Likes Polygons

对于一个给定的多边形 P，我们希望最大化选取的“好割线”的数量。定义在多边形内，水平或竖直的连接两个凹点的线为“**好对角线**”。

定理 4

对于任意的有孔多边形 P，如果 $\mathcal{R}$ 是**最小**矩形剖分，那么

$$|\mathcal{R}| = \frac{1}{4}\left(|P| + 2|C| - 4n\right), \tag{9}$$

其中，C 表示凹点的集合，n 是在 P 内最大的不相交的好对角线个数。

证明.

在最小剖分 $\mathcal{R}$ 中，引理 2 和引理 3 成立，所以最小化 $\mathcal{R}$ 的大小就是最大化割线中不相交的好割线的数量。
我们可以通过选择一组互不相交的好对角线来形成一组合法的好割线，对于不属于任何所选对角线的凹端点，我们可以垂直或水平切割它们。因此，选择最大的不相交的好对角线就形成了最小的矩形划分。

C. Lili Likes Polygons

根据定理 4，本题的其余部分就是找到最大的 N。

将好对角线抽象为节点，两条对角线相交就连一条边连接两个节点。寻找最大不相交的好对角线集合就是找这个图的最大独立集。但是这是一个 NPC 问题。
不过由于对角线只能是水平或者竖直的，因此实际上这张图是一个二分图。

在二分图上的最大独立集可以通过最小顶点覆盖求得，因为任何图中顶点覆盖的补集都是独立集。(| 最大独立集 |= 总点数-| 最小点覆盖 |)

同时，再根据引理 5，| 最大独立集 |= 总点数-| 最小点覆盖 |= 总点数-| 最大匹配 |。

引理 5 (König's Theorem)

在任意二分图中，最大匹配中的边数等于最小顶点覆盖中的顶点数。

C. Lili Likes Polygons

因此，我们需要将这题的好对角线的二分图构建出来。

由于输入的矩形个数 $k \leq 300$，因此离散化后，坐标值范围是 $O(k)$ 的，因此可以直接开一个二维数组进行模拟。

找到所有凸点和凹点的坐标后，直接 $O(|P|^2)$ 枚举凹点对，再 $O(k)$ check。由于每个坐标最多只会属于一个横对角线或者竖对角线，因此均摊下来，所有的 check 是 $O(k^2)$。

之后再枚举竖对角线和横对角线，再检查它们是否相交。由于每个凹点只会属于 $O(1)$ 条横对角线或者竖对角线。因此，这部分枚举的复杂度是 $O(|P|^2)$。

最后，使用 `dinic` 跑二分图最大匹配，时间复杂度是 $O(|P|^{2.5})$。

综上总复杂度 $O(k^2 + |P|^{2.5})$。题目约定 $|P| \leq 2000$，并且实现上常数很小，可以在时限内通过此题。

谢谢大家