博弈论与计算几何

张昕渊

November 3, 2023

公平组合游戏与有向图游戏

- 我们这里仅仅介绍公平组合游戏：
  - 公平：两个人A/B轮流执行操作。当前执行操作玩家的胜负只和当前状态有关，和当前玩家是A/B无关；
  - 信息完全公开；
  - 执行轮数有限。
- 满足上述三个条件的游戏事实上一定可以建模成有向无环图上的行走游戏:
  - 给定一个有向无环图$G=(V,E)$以及放置在某个顶点u的物品。两个人交替完成以下操作：假设目前物品所在位置为x，将该物品通过x的某条出边移到相邻顶点v。不能执行操作的人为输方。
- 构建有向图如下：顶点为所有状态，假设状态x能通过一步转移到状态y则连有向边$x \to y$。
- 求解每个状态是胜态（若目前处于该状态，当前操作的玩家拥有必胜策略）还是败态：DAG上DP即可。初始状态为所有出度为0的顶点为败态。对于顶点u而言，若所有出邻居均为胜态，则该顶点为败态；否则为胜态。

有向图游戏

- 若考虑的是有向图而非有向无环图?
  - 维护一个队列，队列中元素是顶点u，其所有出邻居均被标记胜负态。对于队列中元素u，若其邻居均为胜态，则u为败态；否则为胜态。
  - 未被标记的顶点为平局。

SG数与SG定理

- 多个有向图游戏的组合：给定一个有向图$G = (V, E)$以及m个分别位于顶点$u_1, u_2, \ldots, u_m$的棋子。每一轮移动一枚棋子，不能移动的玩家为输家。
- 如何判定组合游戏中状态的胜负态?
  - Sprague-Grundy数：$sg(u) = \mathrm{mex}\{sg(v) : v \in N^{\mathrm{out}}(u)\}$（mex定义：第一个未出现的最小非负整数）。
  - SG定理：u为胜态当且仅当$sg(u_1) \oplus sg(u_2) \oplus \ldots \oplus sg(u_m) \neq 0$。
  - 证明：假设当前异或为0，则无论如何操作，异或都将变为非零（否则将出现$u \to v$但$sg(u) = sg(v)$的矛盾）。否则，令w为异或和，其写成二进制有k位(0-base)，则一定存在棋子使得其所在节点u满足$sg(u)$在二进制下的第k位为1，因此$sg(u) \oplus w < sg(u)$，即根据mex定义，一定可以找到一个出邻居使其sg数为$sg(u) \oplus w$。

Nim游戏

- Nim游戏：n堆石子，第i堆a_i个，每一个人交替选择一堆石子并拿走若干个（不能不拿）。不能拿的玩家为输家。
- 有向图为$G = (\mathbb{N}, E)$，若$i < j$，则连一条$j \to i$的有向边。$sg(i) = i$。
- 先手必胜当且仅当$a_1 \oplus a_2 \oplus \ldots \oplus a_n \neq 0$。
- Nim游戏的一些变种：
    - 选取石子个数只能为1到K个:有向图为$G = (\mathbb{N}, E)$，若$i < j$且$j - i \leq K$，则连一条$j \to i$的有向边。由数学归纳法可证$sg(i) = i \mod (K + 1)$。
    - 可以选不超过K堆：对于每一位，二进制下该位为1的个数恰好为$K + 1$的倍数，此时为必败态，否则必胜。证明方法等同于$K = 1$的证明，这里不赘述。
    - Anti-nim：不能拿的为赢家。此时胜败态分为两种情形：当a_i均为1时，此时n为偶数为胜态；否则，当异或为0时为败态。
    - Staircase Nim：有n个阶梯，每个阶梯上有n个石子。每一次可以将第i层（$i > 1$）阶梯上的若干个石子移动到$i - 1$层。不能移动的人为败者。问题等价于只考虑偶数层的Nim游戏。

博弈论题中一些常见技巧

- 一些简单的策略：策略窃取；
- 转化为已知的游戏；
- 从简单情形着手/打表找规律。

例题：巧克力

- 给定一个$n \times m$的矩形，初始时全为1。两个玩家轮流执行操作，某个玩家每次选择(i, j)使得$a_{i,j} = 1$，将所有$x \le i, y \le j$的位置$a_{i,j}$置为0，最后操作的玩家为输家。判定先手/后手必胜。
- $n, m \le 10^5$。

例题：巧克力

- 除了$n = m = 1$，否则先手必胜。
- 若不然，则先手选取$(1, 1)$，强制变为后手。

例题：取数游戏

- 给一个长度为偶数的整数序列$a = [a_1, a_2 \ldots, a_n]$，且和为奇数。先手后手轮流取数，每次只能从两端选取。选取完所有数后计算两人的取数总和，总和高者获胜。
- $n \le 10^5, a_i \le 10^9$。

例题：取数游戏

- 先手必胜。
- 先手可以决定全选奇数index/全选偶数index位置上的数。
- 如果先手/后手都最大化自己得分，也可以在$\tilde{O}(n)$时间内计算（Looking for a challenge,Termites），但是证明比较复杂。

例题：Game on Plane

- 给一个凸N边形。每一轮操作玩家选择两个顶点，并且用一条边连接这两个顶点，使得连线内部与多边形以及之前的连线不交。
- Alice/Bob交替执行操作，求胜者。
- $N \leq 5000$。

例题：Game on Plane

- 我们只需要计算sg数即可：$sg(3) = 0$，$sg(n) = \mathrm{mex}\{sg(k) \oplus sg(n - k + 2) : 3 \leq k \leq n - 1\}$。

例题：Colouring Game

- 给定一个$1 \times n$的矩阵A。初始时每一个格子中有红蓝两种颜色之一。现在Alice和Bob轮流执行以下操作：
  - 如果是Alice的回合，他可以选择连续两个格子，这两个格子至少包含一个红色格子。然后将这两个格子的颜色染为白色。
  - 如果是Bob的回合，他可以选择连续两个格子，这两个格子至少包含一个蓝色格子。然后将这两个格子的颜色染为白色。
- 不能操作的玩家为败者，求胜者。
- $n \le 5000$。

例题：Colouring Game

- 显然，若当前格子中有红蓝相邻的格子，那么玩家一定会选红蓝相邻的格子而非只包含红白或者蓝白的格子。
- 因此，游戏将分为两个阶段。第一个阶段Alice和Bob轮流选择红蓝相邻的格子，并将其染为白色；第二个阶段Alice和Bob每一轮将恰好一个红色/蓝色格子染为白色，直至一方无法行动为止。
- 注意到第一阶段中每一次操作恰好删除一个红色格子以及一个蓝色格子。因此，当红色格子数目大于蓝色格子数目时，Alice必胜。若小于，则Bob必胜。
- 当个数相等时，问题等价于第一阶段中谁先无法操作谁输。问题转化为如下：
  - 有m堆石子，第i堆石子a_i个。每次操作可以选择一堆石子个数k大于等于2的堆，取走这堆中两个石子并将剩下的石子划分为两堆（一堆可为空）。两人交替操作，无法操作的为败者。
- 我们只需要计算一堆具有m个石子的sg数即可。由SG定理，$sg(m) = \mathrm{mex}\{sg(i) \oplus sg(n-i-2) | 0 \le i \le n-2\}$。

例题：Game on tree 1

- 给定一个树。初始时有一枚棋子在1号节点。两个人交替完成以下操作：移动棋子到某个其他节点，满足这次移动比上一次移动的距离更大。无法移动的人为输家。
- 树的节点个数不超过10^5。

例题：Game on tree 1

- 后手必胜当且仅当1在直径的中点。
- 必胜方每一次将顶点挪到当前位置关于直径中点的某个对称位置即可，败者一定会远离直径中点。

例题：Game on Tree 2

- 给定一个根为1的有根树。每次操作选择一个非根顶点，将该顶点的子树删除。Alice/Bob交替执行操作，无法操作的为败者。求胜者。
- $n \leq 10^5$。

例题：Game on Tree 2

- 注意到，假设1号节点有k个孩子，我们可以将树拆分为k颗树，每棵树T_i是根与其第i个孩子为根的子树T_i'的并，则问题等价于说有k颗独立的树，每次选一颗树某一个非根顶点。
- 对于这种多个相同结构问题组合的游戏，我们通常考虑计算sg数。此时$sg(T) = \oplus_{i=1}^{d} sg(T_i)$。
- 这已经非常类似于树dp问题了，我们只需要考虑T_i与T_i'的sg数关系即可得到答案。
- 可以利用数学归纳法证明，$sg(T_i) = sg(T_i') + 1$。

例题：mod M game

- 给定$2n$个数$A_1, A_2, \ldots, A_{2n}$，Alice和Bob轮流拿数。若最终Alice选取数的和模M等于Bob的和模M，则Bob获胜；否则Alice获胜。求胜者。
- $n \leq 2 \cdot 10^5$。

例题：mod M game

- 配对思想。
- 令S为总和，A为Alice拿的总和，则Bob获胜当且仅当$2A \equiv S \mod M$。
- 当M为奇数时，Bob只有一个获胜条件：最终和为$2^{-1} \cdot S$。我们将过程逆向考虑，假设Alice最终有两个不相同的数，她一定可以选一个数使得最终和不等于目标值。因此，最后两个数一定要相同，这时候产生一个配对。依次类推，Bob获胜当且仅当每个数均出现偶数次。
- 当M为偶数时，Bob可能有两个获胜条件。同样的逆向考虑，只有当Alice最终剩下两个相同的数，或者两个数模M下相差$M/2$，才可能输。依次类推，我们可以考虑将相同的数/相差为$M/2$的数两两配对。不难证明，当能两两配对且相差为$M/2$的对数为偶数个时Bob获胜；否则Alice获胜。

二维计算几何

- 模板库：https://github.com/kth-competitive-programming/kactl/blob/main/content/geometry
- 向量：$v = (x, y) \in \mathbb{R}^2$：
  - 向量的基本运算：加减、乘某个常数。
  - 向量内积：$v_1 \cdot v_2 = |v_1||v_2|\cos\theta = x_1y_1 + x_2y_2$。可以用于计算夹角。
  - 向量叉积：$v_1 \times v_2 = |v_1||v_2|\sin\theta = x_1y_2 - x_2y_1$。绝对值的两倍代表张成的平行四边形面积，叉积正负代表旋转方向。
  - 向量的旋转：$v = (x, y)$，（逆时针）旋转θ得到向量$P_\theta v$，其中$P_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$

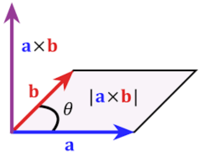

Figure: 叉积

二维计算几何

- 简单应用：求两条直线的交点。
- 我们会有很多种直线的表示方式。我们这里采取的是点+方向向量的表示方法，即直线上的点可由$A + tB$（$t \in \mathbb{R}$）表示，A, B为向量。
- 直线$l_1 : A + tB$，直线$l_2 : C + tD$。
- 注意到直线l_2等价于所有满足$D \times (P - C) = 0$的点P。
- 则$D \times (A + tB - C) = 0$，解得$t = \frac{D \times (A - C)}{D \times B}$。
- 简单思考题：如果给定的是直线上的两点，如何转化为上述表示？

二维计算几何

- 较为复杂的应用：检测两条线段是否有交。
- 一个比较直接的做法：分类讨论线段是否平行，如果非平行算交点。（如何检测平行：方向向量夹角为0）
- 上述方法的劣势：算交点会引入浮点数，很可能会造成精度误差。
- 一个更普遍的做法：
  - 若两条直线平行，存在交点当且仅当两条线段在x轴与y轴的投影均有交。
  - 否则，存在交点当且仅当线段1的两个端点在线段2的异侧且线段2的两个端点在线段1的异侧。
  - 判定点P在包含点A，B的直线哪一侧：看$(A-B)\times(P-B)$的符号（有很多种等价版本）。

二维计算几何

- 极角排序：给定一系列向量，将这一系列向量按照其极角排序。
- 方法1：利用atan2函数计算夹角值，而后进行排序。（缺点：涉及到浮点数就很可能会有精度误差）
- 方法2：通过叉积比较两个向量的夹角大小（需要先确认象限）。

```
sort(range(ls),[&](P u,P v){
        bool bh1=u<make_pair(0ll,0ll);
        bool bh2=v<make_pair(0ll,0ll);
        if (bh1!=bh2) return bh1<bh2;
        return cross(u,v)>0;
});
```

Figure: 代码

二维计算几何

- 两个圆的内/外公切线。

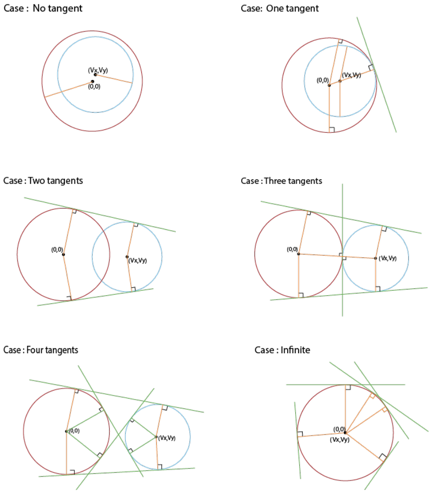

Figure: 内/外公切线

二维计算几何

- 简单多边形：边不自交的多边形。
- 二维简单多边形的面积：
- 任意选取一个顶点P，计算$\left|\frac{1}{2}\sum_{i=0}^{n-1}(P-A_i)\times(P-A_{i+1})\right|$。

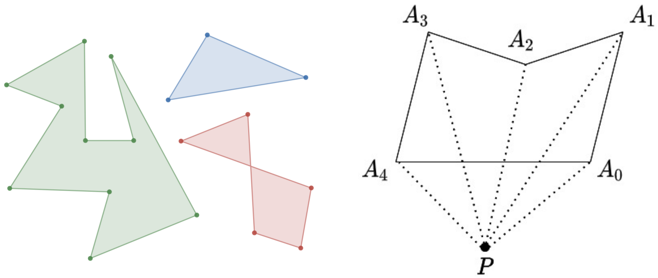

Figure: 简单多边形面积计算

二维计算几何

- 如何判定一个点是否在简单多边形内部。
- 从这个点任意做一条射线，若射线不穿过任何多边形顶点且与简单多边形的交点个数为奇数，则为内部；否则为外部。（时间复杂度：$O(n)$）

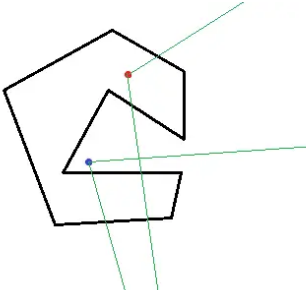

Figure: 射线法判定内部/外部

二维计算几何

- 凸多边形：凸多边形内部/边界任意两点的连线都在凸多边形内部/边界。
- 如何判定一个点在凸多边形内部：选定某个顶点A_0，从A_0出发的射线A_0A_i将整个平面划分为若干个区域，二分查询p落入哪个区域，而后判定p是否在这个区域所在的三角形中即可。
- 判定顶点是否在三角形中：可以用叉积算面积。

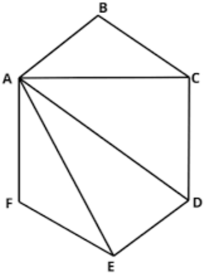

Figure: 凸多边形内部

二维计算几何

- 类似的二分可以找到一个外部点的两条" 切线" 。

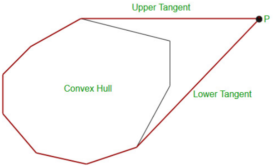

Figure: 切线

二维计算几何：最近点对问题

- 我们这里介绍随机增量的做法。分治法可以在OI-wiki/各种博客上找到。
- 我们将所有点随机重排，随后按顺序插入点，维护前i个点两两之间的最小距离，假设为d_i。
- 假设当前最小距离为d_i，我们将整个坐标划分为$d_i/2 \times d_i/2$的网格。每个格子中最多有一个点。
- 假设现在插入一个新的顶点x_{i+1}，我们只需要查看x_{i+1}所在格子周围常数个格子，更新答案即可。如果最小距离变小，则重构网格。
- 最差时间复杂度$O(n^2)$（每插入一个点都需要重构）。但是，注意到插入第i个点，答案被更新的概率是$\frac{2}{i}$。因此，期望时间为线性。

二维计算几何：凸包

- 给定n个顶点，凸包(convex hull)指的是包含这n个顶点的最小凸多边形。

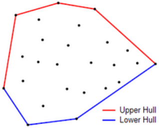

Figure: 凸包算法

- 凸包算法：将所有点按x作为第一关键字、y作为第二关键字进行排序。而后利用单调栈求得上凸壳、下凸壳即可。

二维计算几何：最远点对

- 最远点对肯定在凸包上。
- 对该凸多边形应用旋转卡壳算法（i的最大值点是$f(i)$，满足f单调，$f(n) - f(0) = n$）。

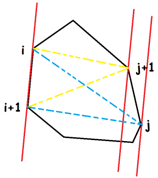

Figure: 旋转卡壳算法

- 旋转卡壳还可以解决最小矩形覆盖的问题。

例题：Keep the Parade Safe

- 给定若干个蓝点以及绿点，求满足下列条件的绿点个数。
    - 存在四个蓝点组成一个非退化的四边形包含该绿点。
- 绿/蓝点个数不超过1000。

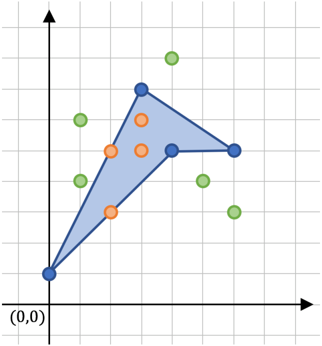

Figure: 图示：橙色点是满足条件的点

例题：Keep the Parade Safe

- 若蓝点的凸包顶点个数大于等于4，则答案为凸包内部的点。
- 若顶点个数等于3，且严格包含某个蓝点，则答案也为凸包内的点。
- 否则，答案为0。

例题：Illumination

- 给定一个不透光的凸n-边形以及凸多边形外的若干个点光源，选择尽可能少的点光源，使得凸多边形全部被照亮。
- $n \leq 10^5$，点光源个数不超过10^5个。

例题：Illumination

- 每个点光源的照亮区域如下图所示。

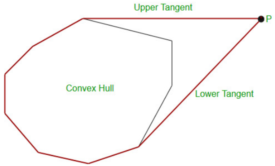

Figure: 切线

- 对每个点光源，能够照亮的区域是一段区间，问题转化为若干个圆上的区间，选最少个数使得覆盖整个圆周。
- 这是一个经典的贪心，可以通过倍增在$O(n \log n)$时间内解决。

例题：Line distance

- 给定n个两两不同的二维坐标点P_i。过任意两点得到一条直线ℓ，原点到这条直线ℓ的距离为d_ℓ。求所有d_ℓ中第K大的值。
- $n \leq 10^5$，7s。

例题：Line distance

- 几何题中一个常见思想：二分答案。
- 问题转化为给定一个圆，求有多少点对满足过该点对的直线与圆相交。
- 不妨假设所有点都在圆外。不相交当且仅当共切点交错排列。
- 沿着圆扫一圈可以在$O(n \log n \log R/\epsilon)$的时间内求得答案。

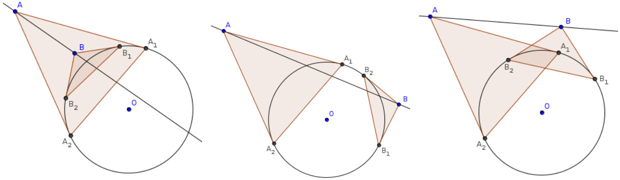

Figure: 不同的情形

二维计算几何：半平面交

- 给定若干个半平面：$Ax + By + C \le 0$，求这一系列半平面的交的面积/是否为空。
- 有一个比较常见的做法是S&I算法。但是我们今天介绍一个基于三分的做法。
- 不妨假设不存在竖直的半平面。这样的话每一个半平面都将是$y \le ax + b$或者$y \ge ax + b$的形式。
- 假设我们只考虑第一种，交出来的区域如下图。是一个上凸壳f（concave function）。（构造这个上凸壳只需要按斜率排序，维护一个单调栈即可）。

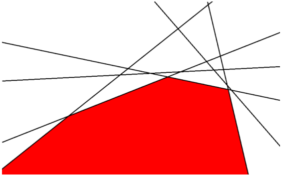

Figure: 上凸壳

二维计算几何：半平面交

- 假设我们只考虑第一种，交出来的区域如下图。是一个上凸壳f（concave function）。（构造这个上凸壳只需要按斜率排序，维护一个单调栈即可）。
- 只考虑第二种，交出来的区域是一个下凸壳g（convex function）。
- 半平面交非空当且仅当存在$x \in \mathbb{R}$满足$g(x) \leq f(x)$。注意到$g - f$还是一个凸函数(convex function)，因此我们可以三分求出最小值。
- 如果需要求面积，我们只需要找到最小值后二分找到两端满足$g(x_i) = f(x_i)$的点x_i。由于函数是一个线性函数，我们可以逐段求面积。

例题：不对称值

- 给定一个凸n-边形P。对于多边形内顶点O而言，其不对称值为$\max_{\substack{X,Y\in\partial P\\ X,O,Y\text{共线}}} \frac{|OX|}{|OY|}$，其中$\partial P$为$P$的边界。
- 多边形P的不对称值为内部点不对称值的最小值。
- 给定一个凸多边形，求其不对称值α。
- $n \leq 1000$。

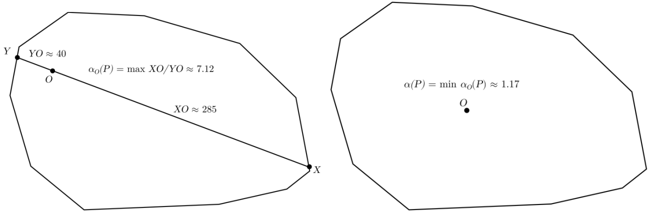

Figure: 图示

例题：不对称值

- 对于凸多边形P以及给定点O而言，其不对称值小于等于α当且仅当对于任意的$X \in \partial P$，顶点Y满足$\vec{OX} = -\alpha \vec{OY}$且$Y$在$P$内部或边界上。
- 注意到，对于任意的$X \in \partial P$，顶点Y的轨迹其实本质上是凸多边形镜像缩放到原来的$1/\alpha$后经过平移所得到。
- 题目中需要求最小的α，即最小（最小幅度改变）的缩放比例使得凸多边形镜像缩放后可以放入到原多边形中。（可以证明这是充要的）
- 因此，我们可以对α二分，问题立刻变成了判定一个凸多边形是否可以通过平移放入另一个凸多边形内部。
- 能放入当且仅当每一个顶点都在多边形内部，因此，我们只需要枚举平移量，列出若干个方程判定是否有解即可。注意到方程均是$ax + by + c \le 0$的形式，因此转化为半平面交问题。
- Remark：上述做法是$O(n^2 \log n \log R/\epsilon)$的复杂度，$n^2$是由于半平面交的限制有$O(n^2)$个。但其实只需要$O(n)$个即可。

例题：不对称值

- 对于凸多边形P以及给定点O而言，其不对称值小于等于α当且仅当对于任意的$X \in \partial P$，顶点Y满足$\vec{OX} = -\alpha\vec{OY}$且$Y$在$P$内部或边界上。
- 注意到，对于任意的$X \in \partial P$，顶点Y的轨迹其实本质上是凸多边形镜像缩放到原来的$1/\alpha$后经过平移所得到。
- 题目中需要求最小的α，即最小（最小幅度改变）的缩放比例使得凸多边形镜像缩放后可以放入到原多边形中。（可以证明这是充要的）
- 因此，我们可以对α二分，问题立刻变成了判定一个凸多边形是否可以通过平移放入另一个凸多边形内部。
- 能放入当且仅当每一个顶点都在多边形内部，因此，我们只需要枚举平移量，列出若干个方程判定是否有解即可。注意到方程均是$ax + by + c \le 0$的形式，因此转化为半平面交问题。
- Remark：上述做法是$O(n^2 \log n \log R/\epsilon)$的复杂度，$n^2$是由于半平面交的限制有$O(n^2)$个。但其实只需要$O(n)$个即可。

Pick定理

- Pick定理：给定一个整点/格点多边形（即多边形的顶点坐标均为整点）。该整点多边形的面积$S = I + \frac{B}{2} - 1$，其中I为严格包括在内部的整点个数，B为在边界上的整点个数。

例题：Gregor and the Odd Cows（Easy version）

- 给定平面上n个整点，求选三个不同点使得这三个点组成的三角形严格包含的整点个数为奇数个。
- $n \le 6000$，坐标x, y均为偶数。

例题：Gregor and the Odd Cows（Easy version）

- 由Pick定理，$2S = 2I + B - 2$。由于坐标均为偶数，因此$S = \frac{1}{2}|x_1y_2 - x_2y_1|$也是偶数。由于我们需要$I$为奇数，两边模4可得$B \equiv 0 \mod 4$。
- 给定两个点(x_1, y_1), (x_2, y_2)，边上的整点个数为$\gcd(|x_1 - x_2|, |y_1 - y_2|) + 1$。
- 三角形的三个端点都被算了两次，三角形边界上的整点个数为$\gcd(|x_1 - x_2|, |y_1 - y_2|) + \gcd(|x_1 - x_3|, |y_1 - y_3|) + \gcd(|x_2 - x_3|, |y_2 - y_3|)$。
- 注意到$\gcd(|x_1 - x_2|, |y_1 - y_2|) \equiv 0 \mod 4$当且仅当$x_1 \equiv x_2 \mod 4$且$y_1 \equiv y_2 \mod 4$。因此我们对顶点模4分类，然后枚举每一类中选择即可。

Minkowski和

- Minkowski和：集合$A + B = \{a + b \mid a \in A, b \in B\}$。
- 两个凸多边形的Minkowski和：
  - 仍然是一个凸多边形；
  - 每一条边由这两个凸多边形的边通过平移得到；
  - $O(n + m\log(n + m))$的做法：将这两个凸多边形的边的方向向量进行极角排序，按照顺序依次添加即得到和的多边形。为了确认位置，只需要确认最小x取值与最小y取值即可。
  - $\log(n + m)$可以去除，原因是这里的合并类似于归并排序中的合并，可以在线性时间内完成。

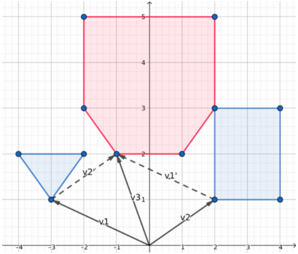

Figure: 闵可夫斯基和

Minkowski和

- Minkowski和：集合$A + B = \{a + b \mid a \in A, b \in B\}$。
- 求两个凸多边形的距离：只需要求$A - B$这个凸多边形离原点的距离即可，$-B$：将多边形根据原点镜像。