计算几何

清华大学 张瑞喆

计算几何是什么

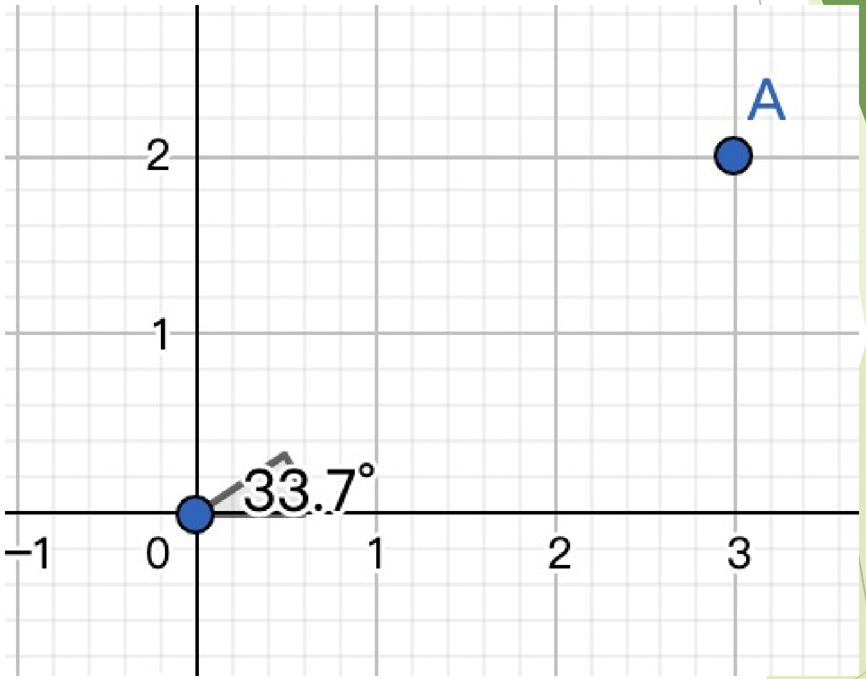
- 编程解决几何问题
- 二维、三维、高维问题
- 几何+编程
  - 几何注重数形结合
  - 编程注重时空效率

本节内容

- 二维基本图形的表示
- 二维基础运算
- 二维进阶算法
  - 凸包
  - 图形的并
  - 旋转卡壳
- 三维和高维几何
- 经典问题

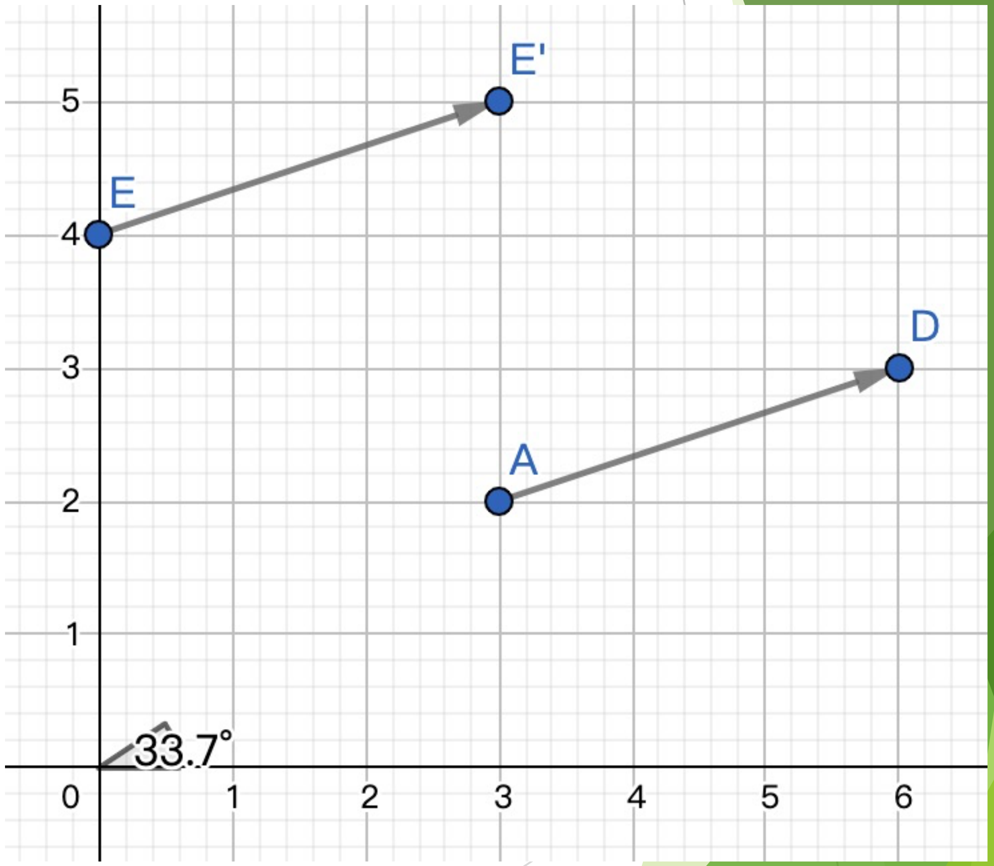
二维图形的基本表示

- 点的表示
  - 平面直角坐标系
  - 极坐标系
- 直线、射线、线段的表示
  - 直线：两点决定
  - 线段：两端点决定
  - 射线：端点和线上一点
- 思考：是否还有其他表示方式?
- 思考：如何表示圆和三角形?



二维图形的基本表示

- 向量
  - 有“方向”的“量”
  - 用$\overrightarrow{AD}$表示从A到D的向量
  - $\overrightarrow{AD} = (D_x - A_x, D_y - A_y)$
- 位置
- 模长
  - $|\overrightarrow{AD}| = \sqrt{(D_x - A_x)^2 + (D_y - A_y)^2}$

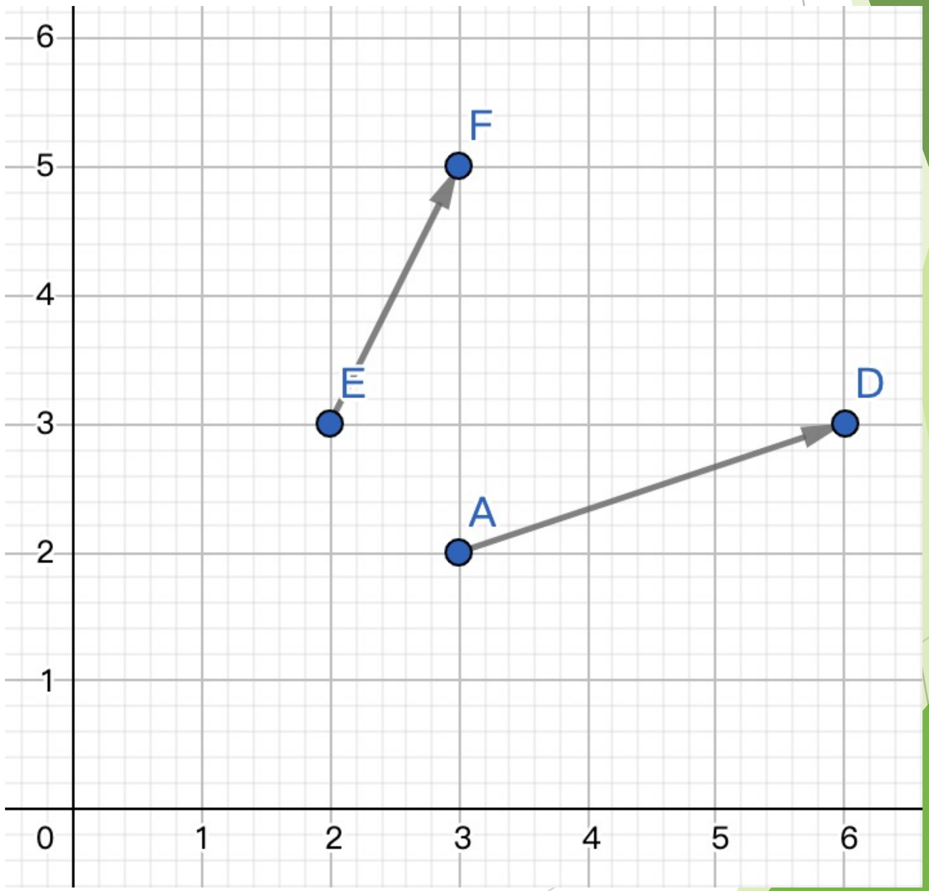


二维基础运算-N则运算

- 加法
- 减法
- 乘除常数
  - 正则化
- 叉积
- 点积

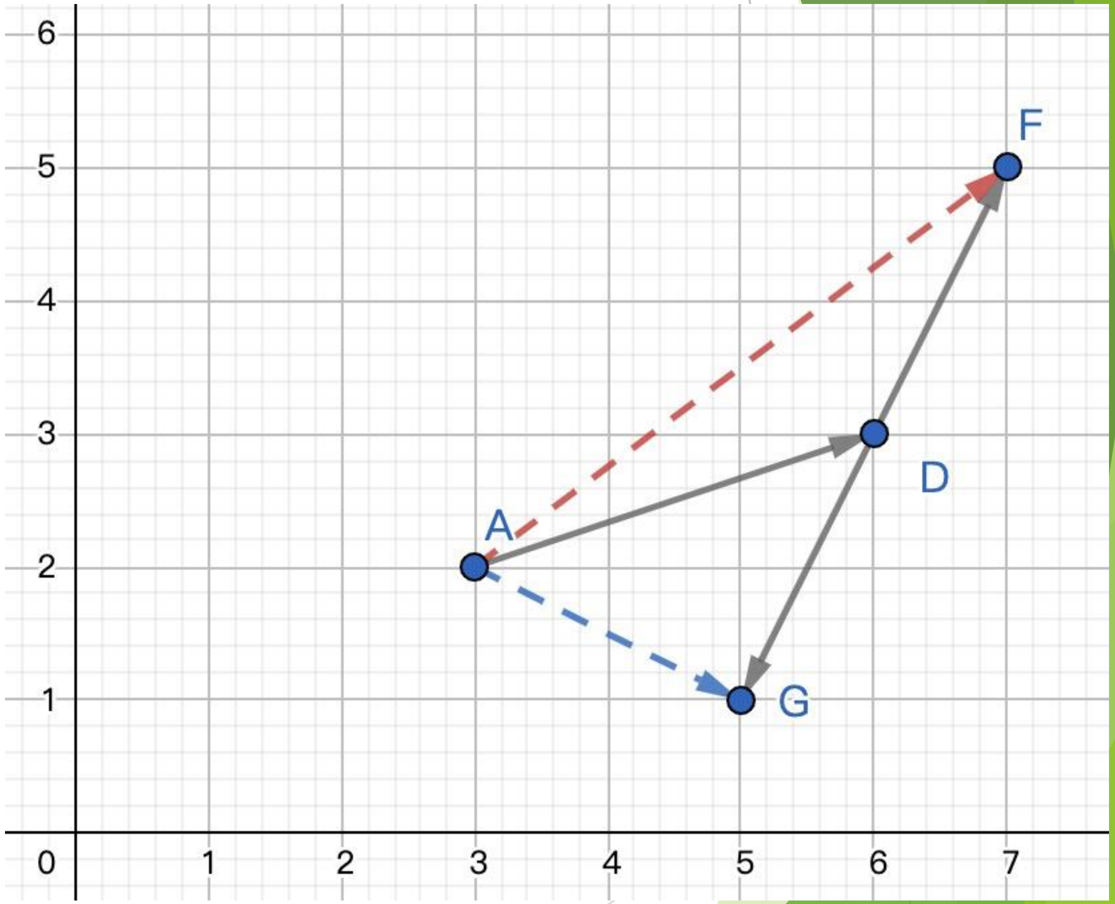
二维基础运算-N则运算

- 加法
  - A= (A_x, A_y) B= (B_x, B_y)
  - A+$B = (A_x + B_x, A_y + B_y)$
- 减法
  - A−$B = (A_x - B_x, A_y - B_y)$
- 乘除常数
  - $kA = (kA_x, kA_y)$



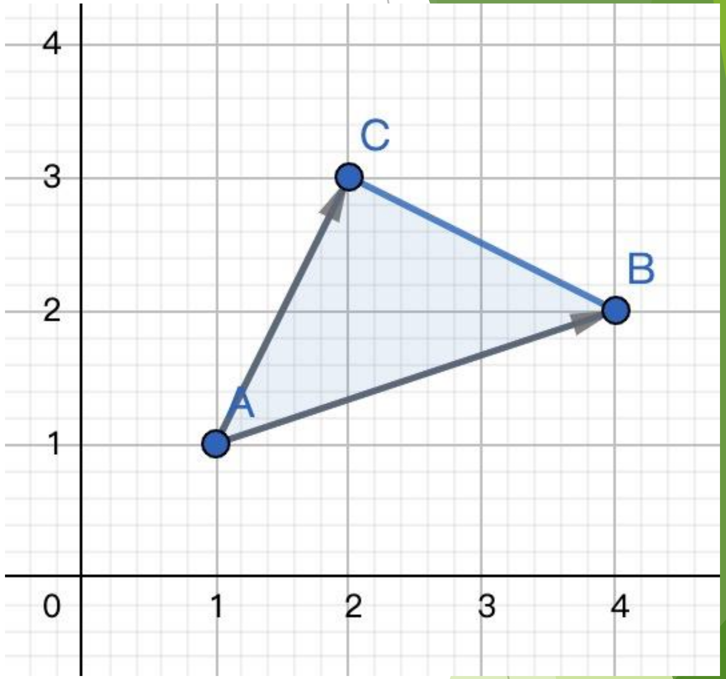
二维基础运算-N则运算

- 加法
  - A= (A_x, A_y) B= (B_x, B_y)
  - A+$B = (A_x + B_x, A_y + B_y)$
- 减法
  - A−$B = (A_x - B_x, A_y - B_y)$
- 乘除常数
  - $kA = (kA_x, kA_y)$



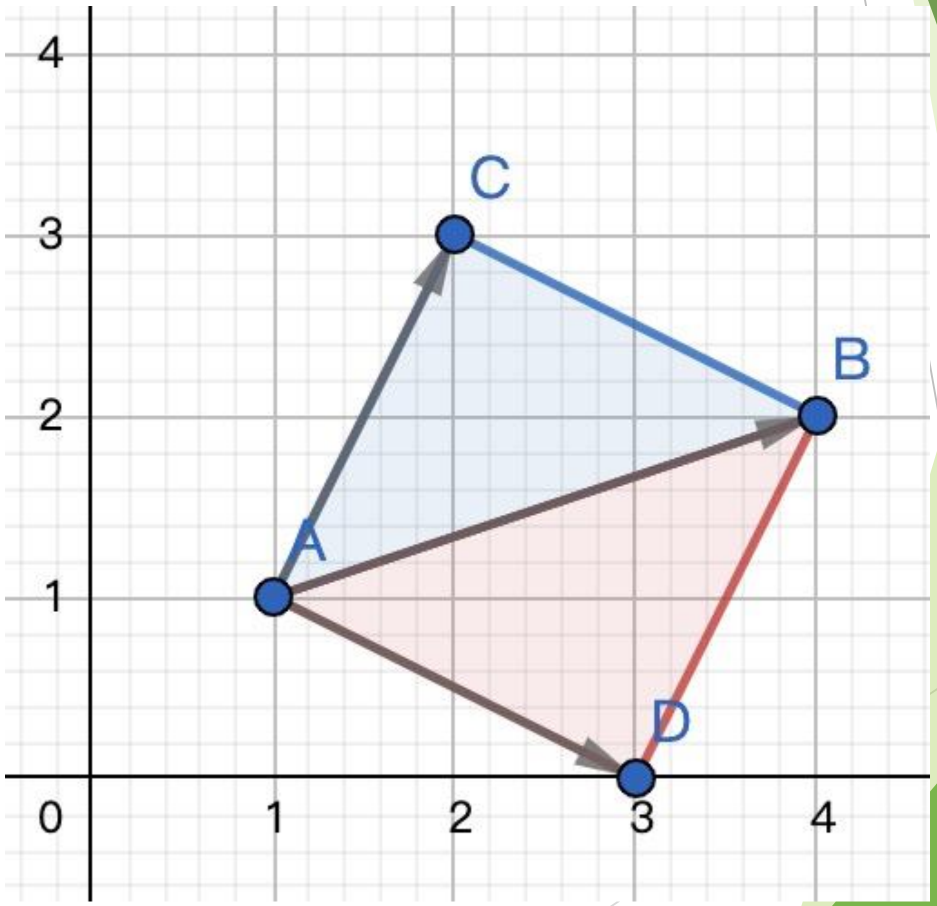
二维基础运算-N则运算

- 叉积
  - $A \times B = A_x B_y - B_x A_y$
  - 意义?
- 点积
  - $A \cdot B = A_x B_x + A_y B_y = |A||B|cos(A, B)$
  - 用途?



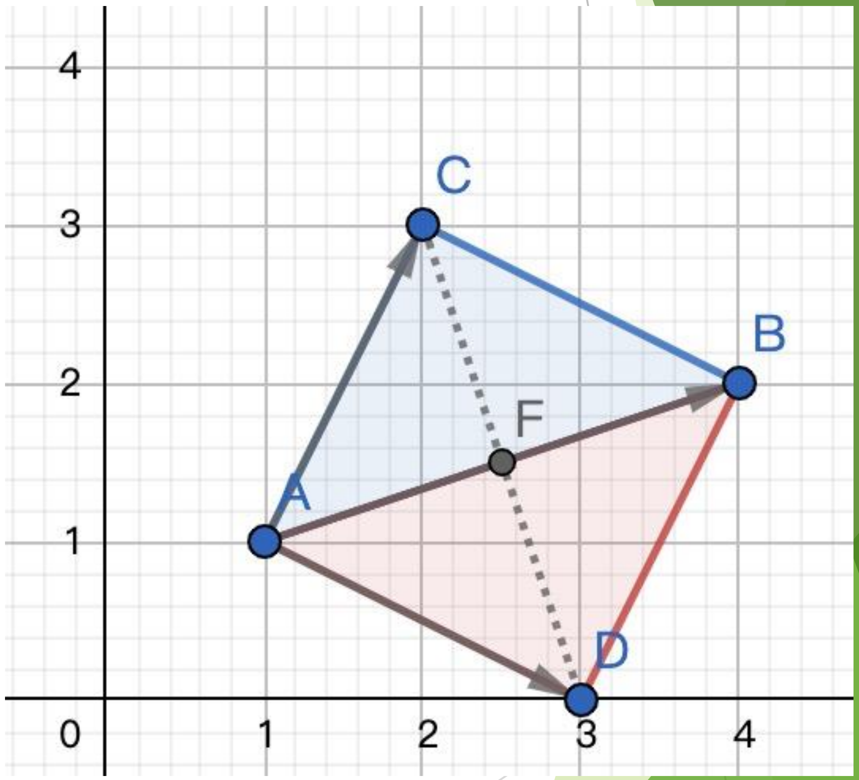
二维基础运算-线段交的判定

- 如何判断线段交?
- 设有线段AB、CD
- $(AB \times AC) \times (AB \times AD) \leq 0$
- 且
- $(\mathrm{CD} \times CA) \times (CD \times CB) \leq 0$
- 问题:
  - 什么时候会取到等号?
  - 为什么要判两组?

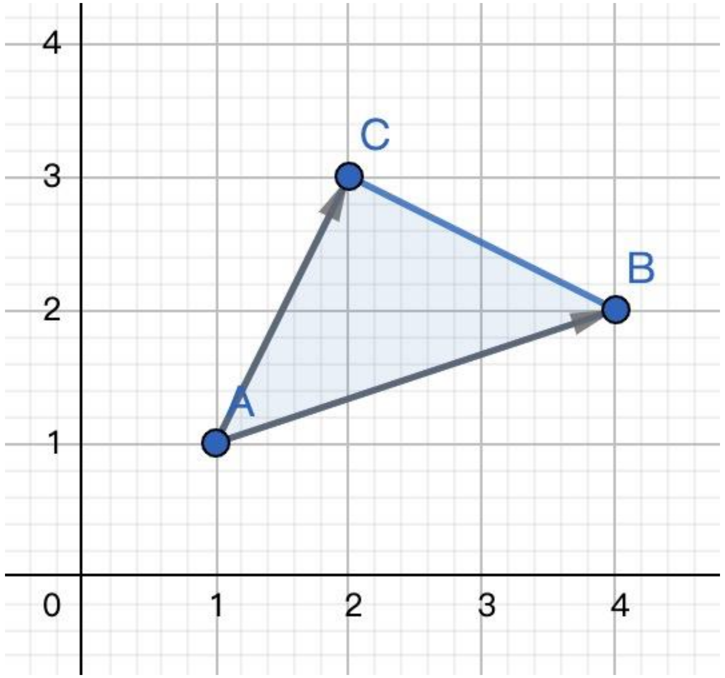


二维基础运算-线段的交点

- 如图，有$|DF|:|CF| = |AB \times AD|:|AB \times AC|$
- 拓展到一般情况，推出：
- $F = D + (C - D) \times \frac{(AD \times AB)}{(AD \times AB + AB \times AC)}$
- 以上公式是否总是可计算?
- 如果线段不相交，求出的是什么?

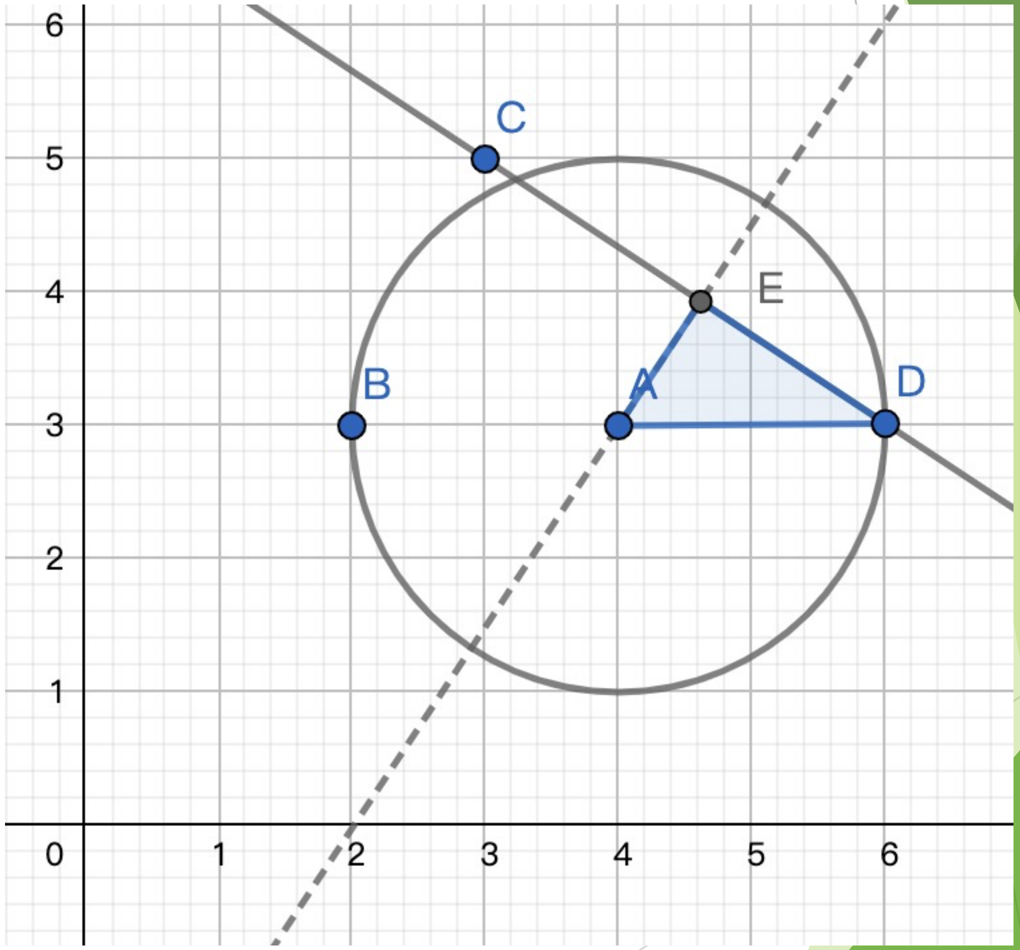


二维基础运算-面积



二维基础运算-圆相关

- 两圆的关系
  - $d > R + r$ 外离
  - $d = R + r$ 外切
  - $|R - r| < d < R + r$ 相交
  - $d = |R - r|$ 内切
  - $d < |R - r|$ 内含
- 圆和直线的交点
  - 先判断距离!



二维基础运算--一些简单的问题

- 点A是否在直线BC上
- 点A是否在线段BC上
- 点A到直线BC的距离
- 点A到线段BC的距离
- 点A到直线BC的垂足坐标
- 两圆交点

二维基础运算-偏序相关

- 如何给平面上的点排序?
  - 方法1、按x,y坐标进行双关键字排序
  - 方法2、按极坐标排序
- 极坐标排序的问题:
  - 大小关系是循环的
  - 强行规定起点
  - cmp(A,B): 先判断是否在上半平面，再判断二者关系

休息一下

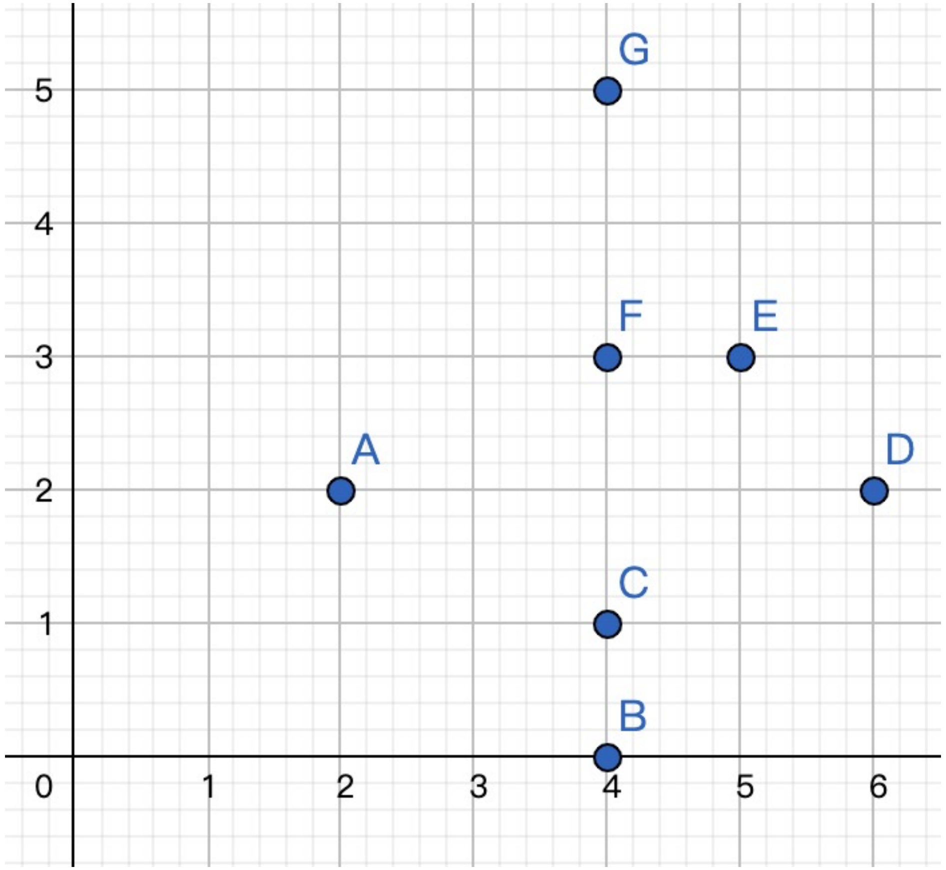
二维进阶算法-凸包

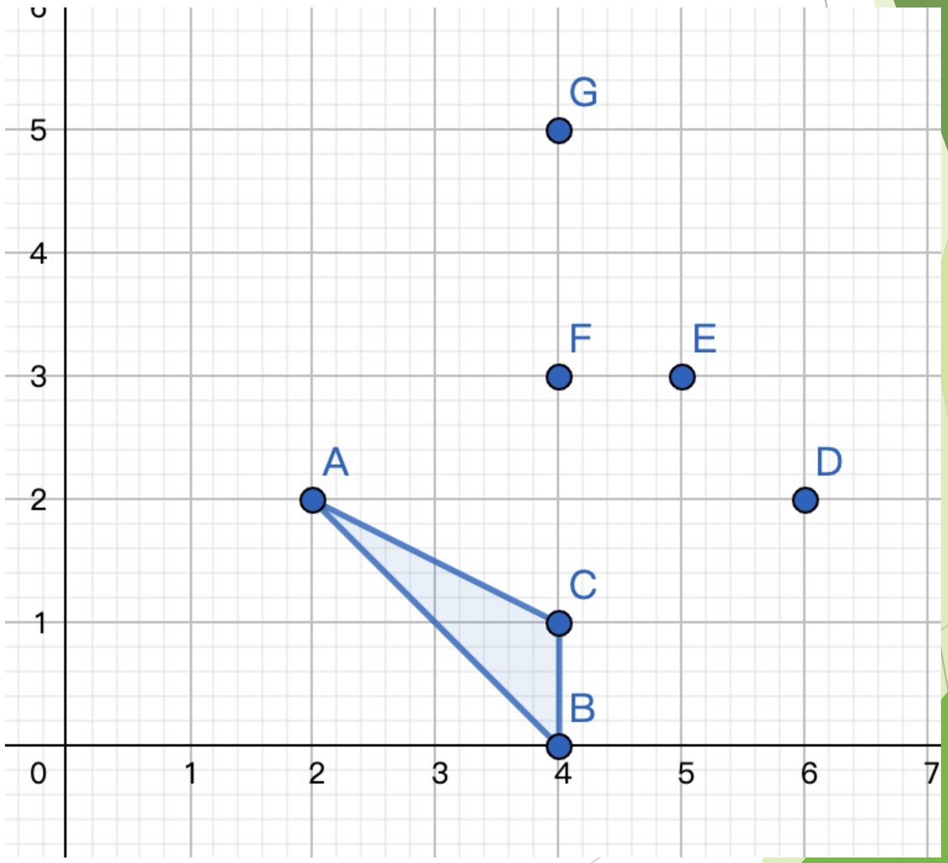
- 凸包问题：
  - 平面上有N个点，求一个【凸多边形】，满足其是面积最小且包含了所有N个点（在内部或边界）
- 一些思考：
  - 是否是唯一的？
  - 是否会退化？

二维进阶算法-凸包

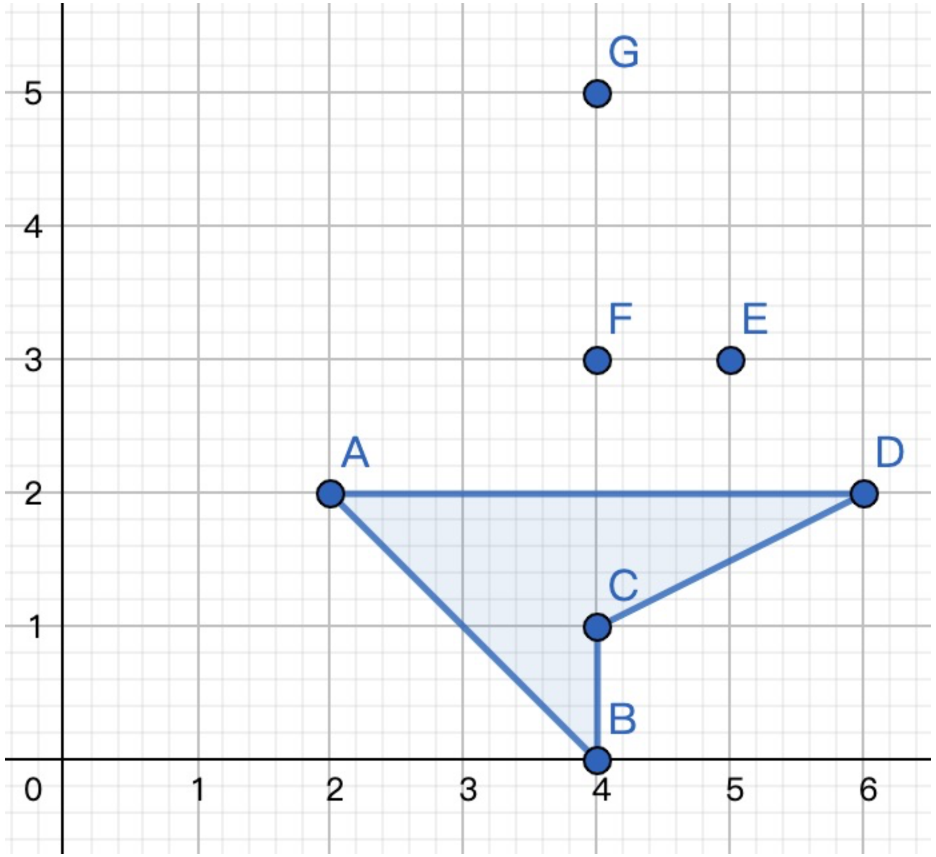
- Graham扫描
- 1、选出x坐标最小（y坐标作为第二关键字）的点A，其必然在凸包上
- 2、将其他点按照向量AX的位置，排出极角顺序
- 3、维护一个栈
- 4、按极角顺序枚举点，判断栈顶的点和当前的点是否会形成矛盾，形成矛盾则将栈顶的点弹出
- 5、将枚举的点插入栈中

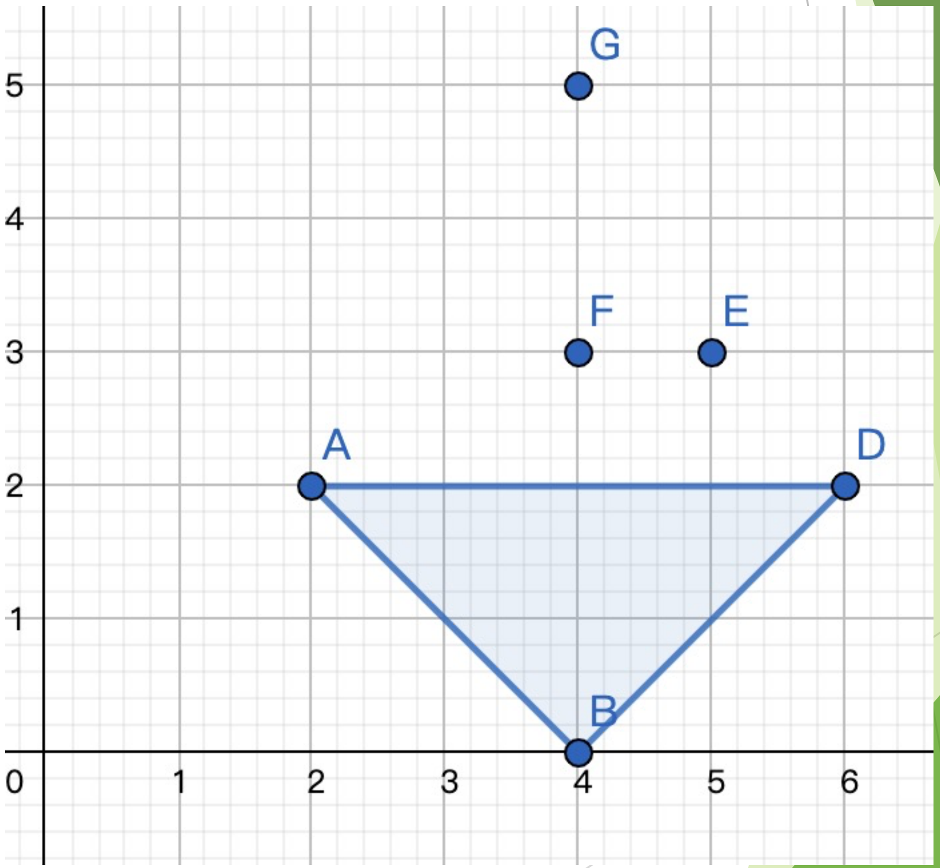
二维进阶算法-凸包



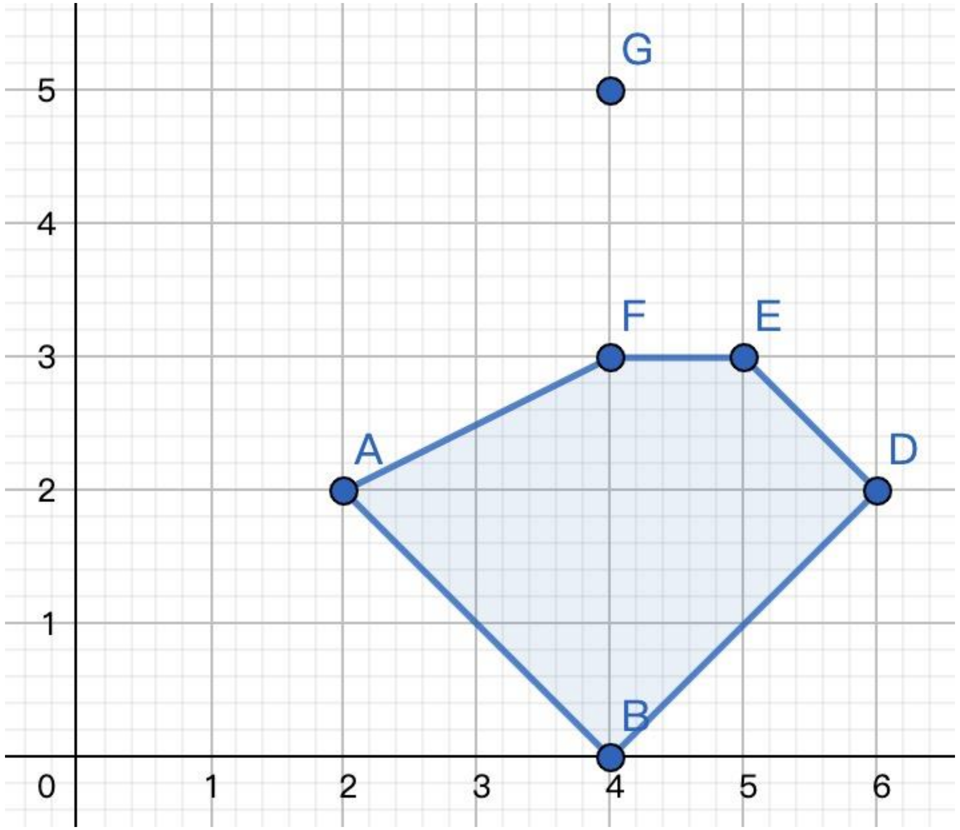


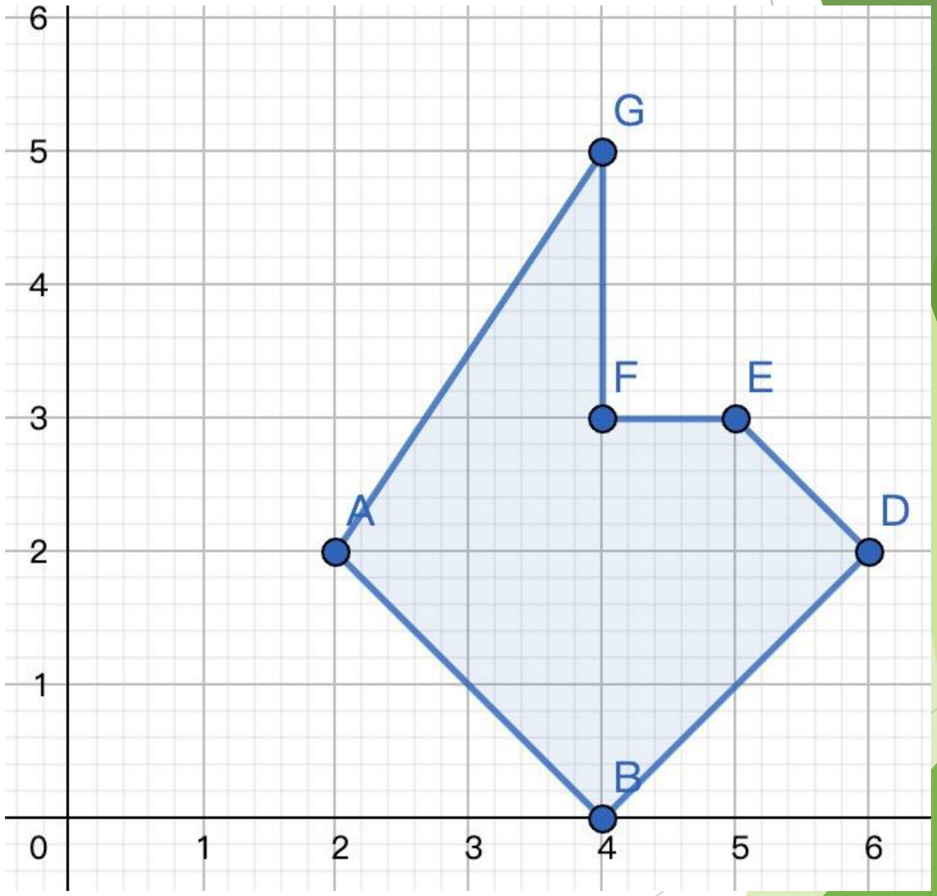
二维进阶算法-凸包



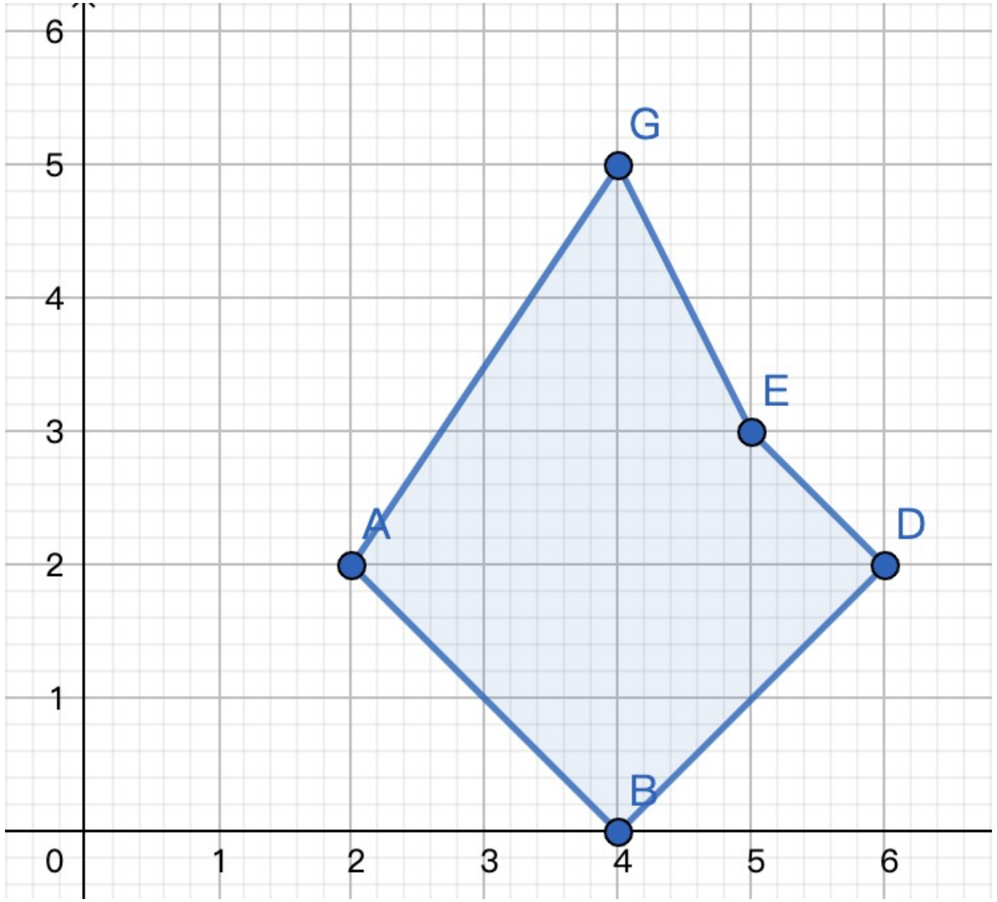


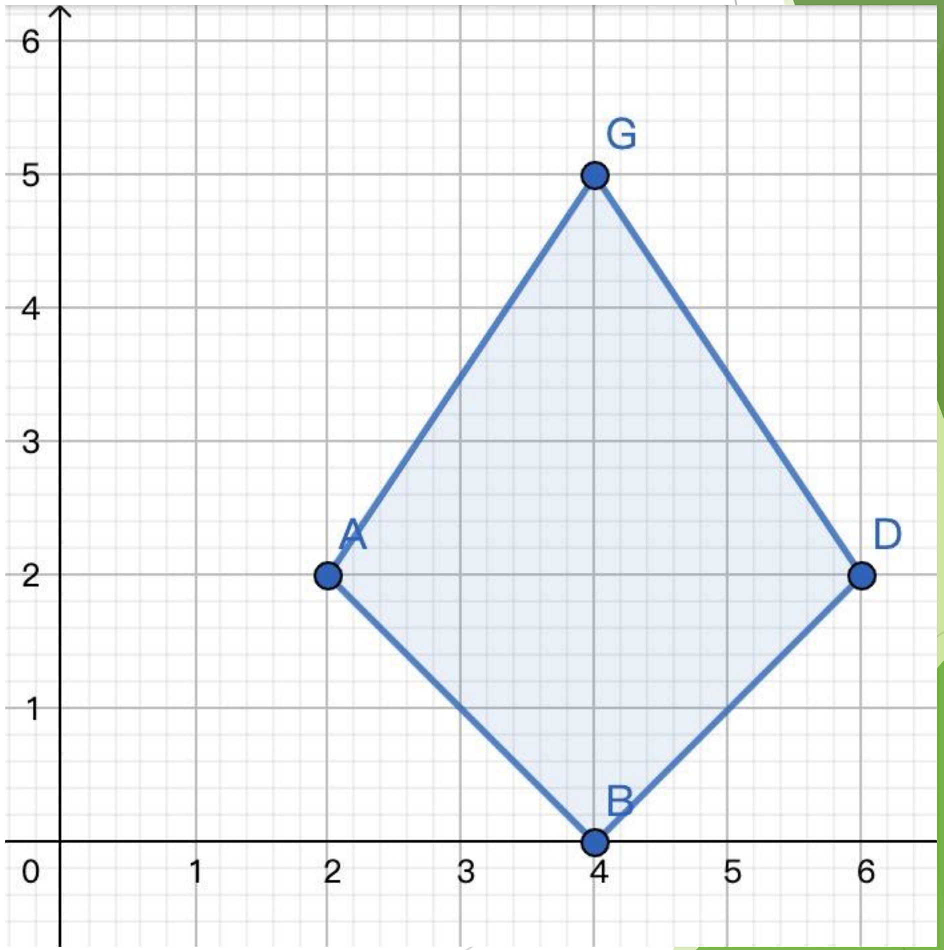
二维进阶算法-凸包





二维进阶算法-凸包





二维进阶算法-凸包

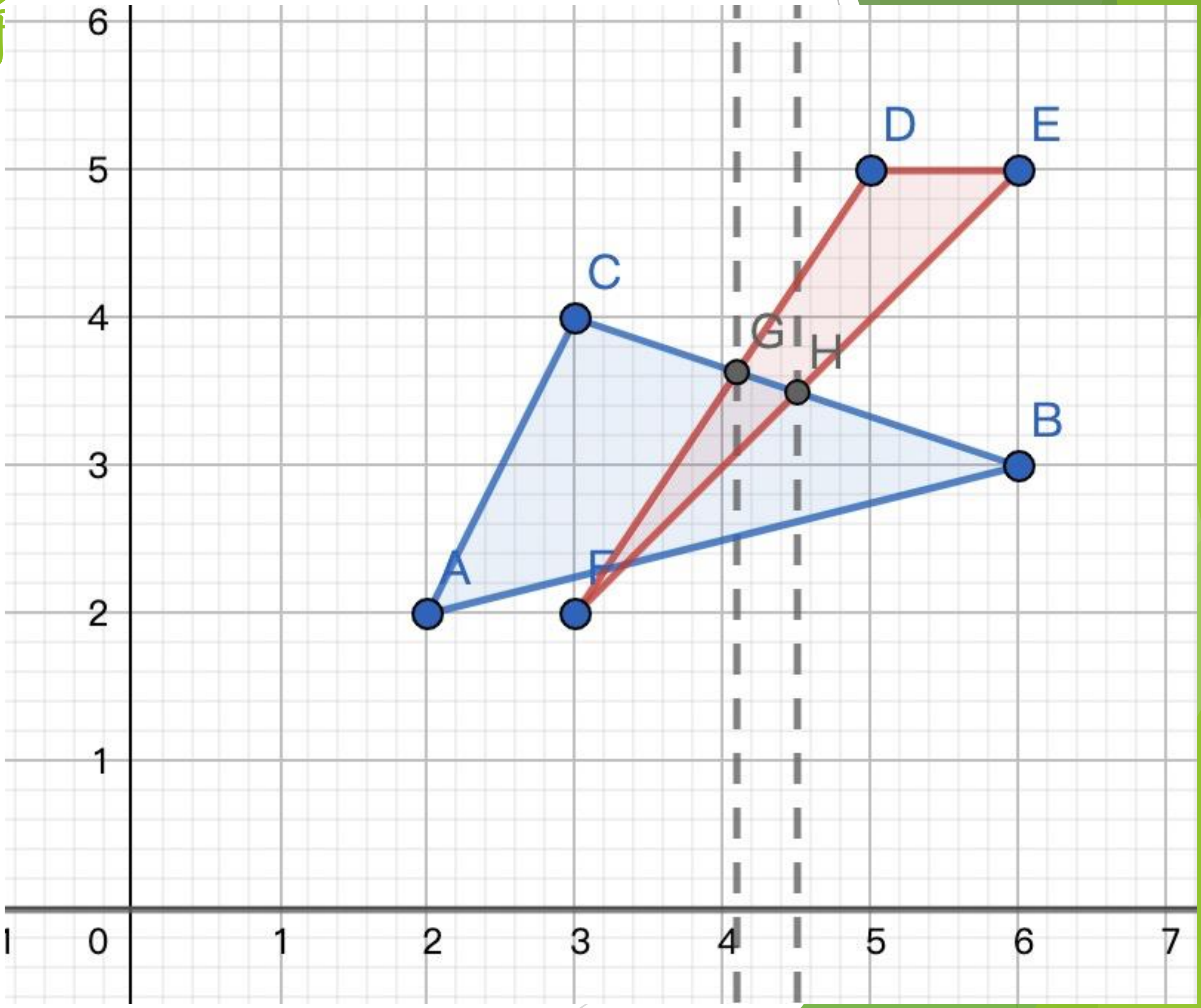
- 注意事项：
  - 要考虑三点共线的情况（推荐不保留多余的点）
- 特殊情况：
  - 退化成点、线
- 其他做法
  - 卷包裹，先把点从左到右排序，从左到右卷出下边界；再反过来卷出上边界，然后合并
  - 随机增量法

二维进阶算法-面积并

- 多边形面积并
  - 给定300个三角形，求覆盖的总面积
- 圆面积并
  - 给定300个圆，求覆盖的总面积
- 复杂图形面积
  - 给定1000个三角形和1000个圆，求覆盖的总面积

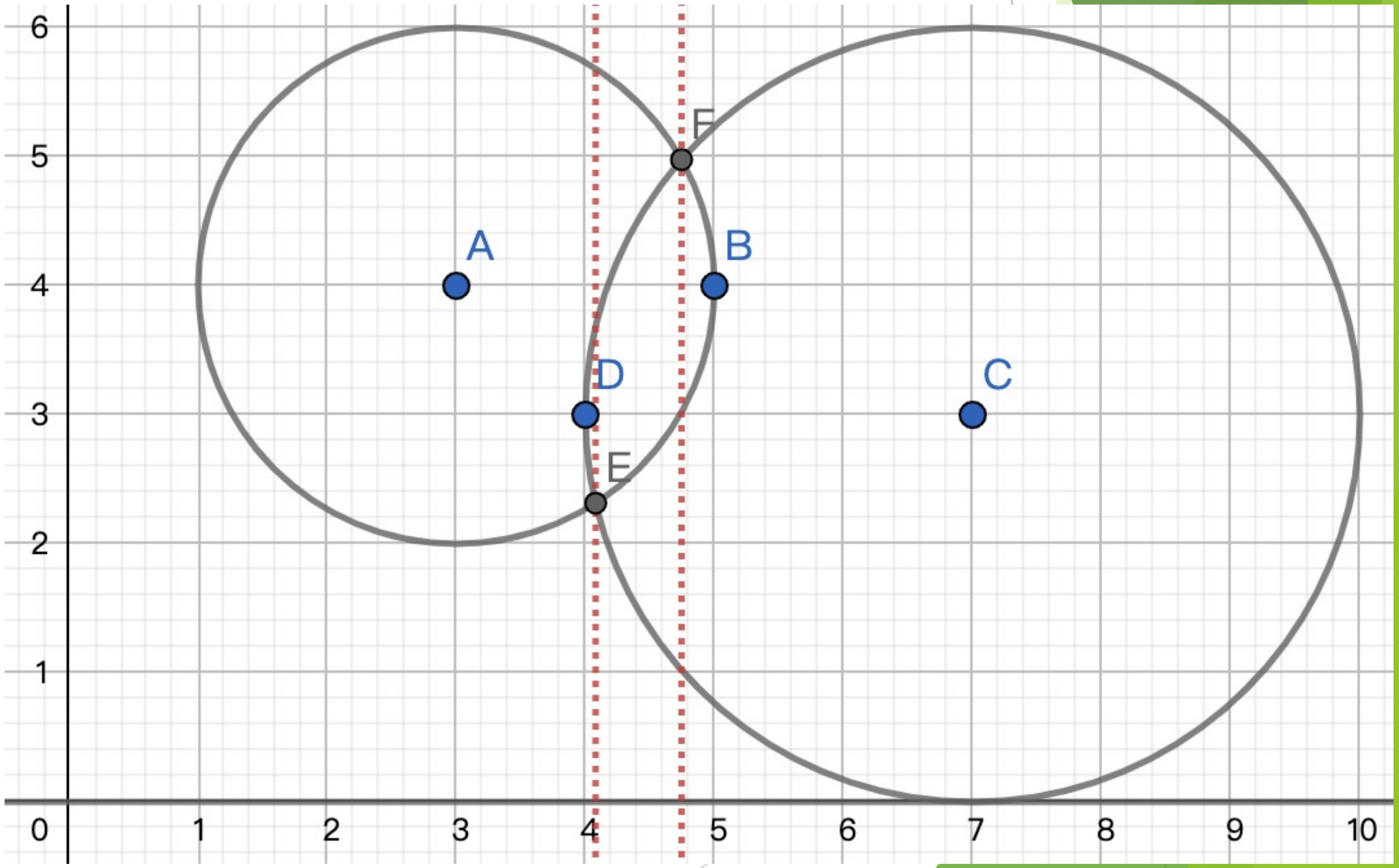
二维进阶算法-面积并

- 1、 先求出三角形的边的交点
- 2、 对交点进行排序
- 3、 相邻2交点之间，面积为若干梯形面积



二维进阶算法-面积并

- 1、先求出圆~~三角形的边~~的交点
- 2、对交点进行排序
- 3、相邻2交点之间，面积为？？？~~若干梯形~~面积

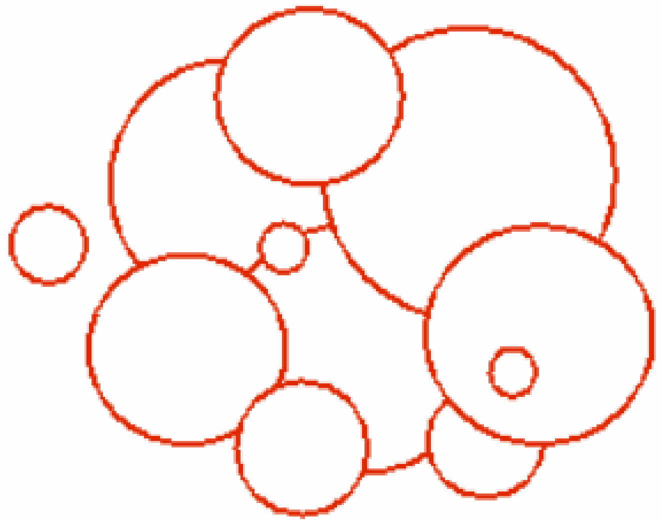


二维进阶算法-自适应的辛普森积分

- 如果有更多的图形，且精度要求不高
- 将平面竖剖成若干区间，对于每个区间内求梯形面积
- 调整1：辛普森积分
  - 如果出现圆等曲边图形，梯形面积往往会低估
  - 用(a+4b+c)/6*h替代梯形面积（其中a、c分别为上下底，b为中间位置的垂线被染色的面积。
- 调整2：自适应
  - 对于一个区间，如果将其切分成两个小区间求解得到的面积和直接求一致，则认为误差可以接受；否则，递归求解

思考题

- 下落的圆盘
  - 有1000个实心圆盘从天而降，后落下的会盖住先落下的
  - 问露出来的【周长】的总长度

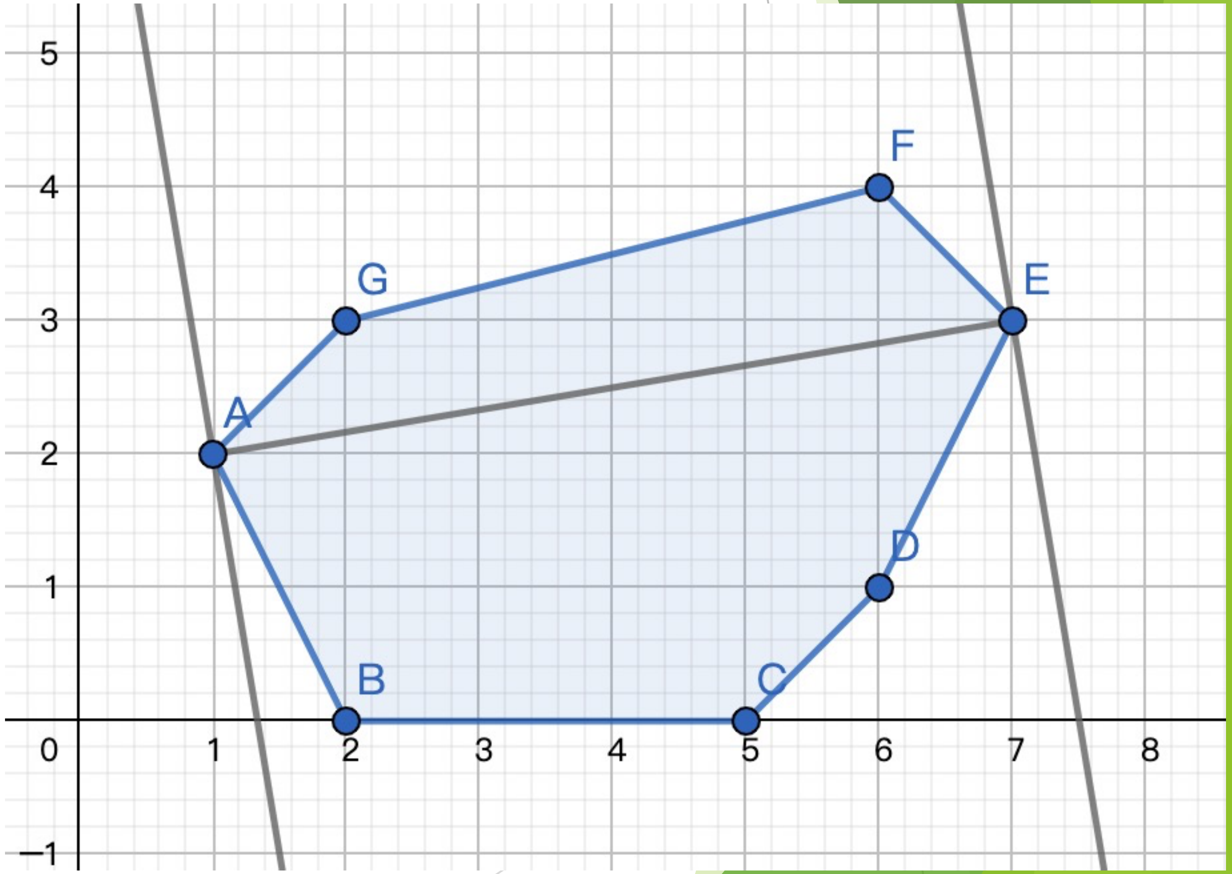


二维进阶算法-旋转卡壳

- 给定平面上的10w个点，求最远点对

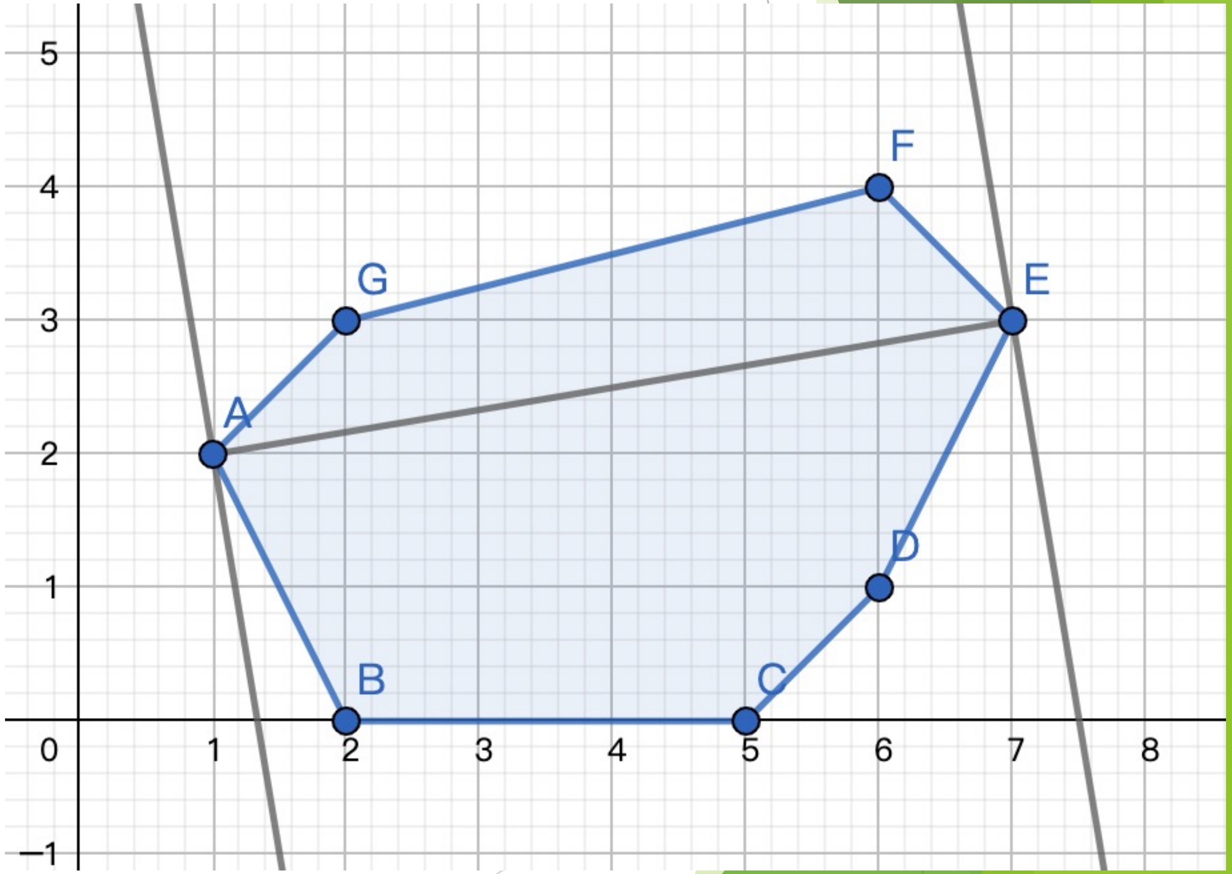
二维进阶算法-旋转卡壳

- 给定平面上的10w个点，求最远点对
- 先构造凸包，最远点对一定在凸包上
- 10w个点？
- 对于最远点对连线做两条垂线，凸包一定在两条垂线之间
- 反过来考虑，有两条直线绕着凸多边形逆时针旋转



二维进阶算法-旋转卡壳

- 逆时针旋转一周时，每个点均摊只会被卡到一次
- 一侧每次往后转一个点（对应了一个斜率区间）
- 另一侧转若干个点（也可能不转）
- O(n)

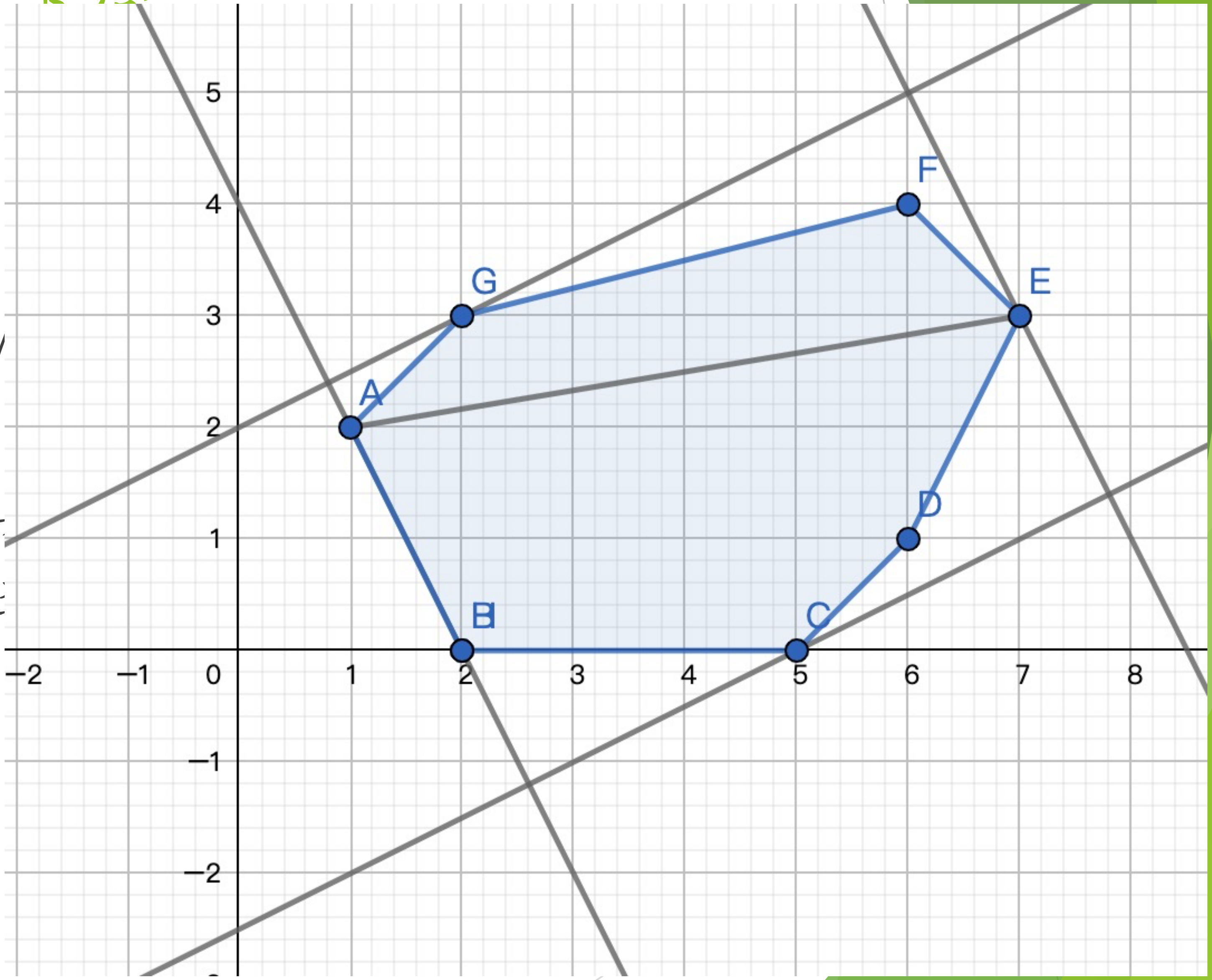


二维进阶算法-旋转卡壳

- 给定平面上的10w个点
- 求面积最小的能覆盖这10w个点的矩形

二维进阶算法-旋转卡壳

- 给定平面上的10w个点
- 求面积最小的能覆盖这10v
- 同样使用旋转卡壳
- 矩形一定有一条边和凸包
- 枚举这条边，卡另外三条

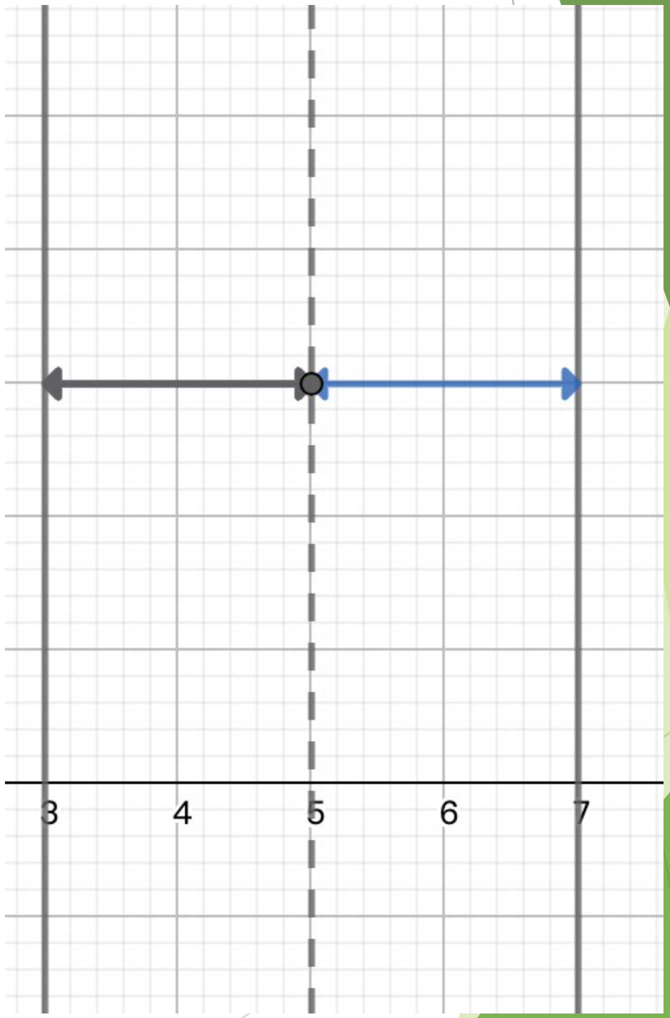


二维进阶算法-最近点对

- 给定10w个点，求最近的一对点
- 好像不能旋转卡壳了。。。？

二维进阶算法-最近点对

- 给定10w个点，求最近的一对点
- 好像不能旋转卡壳了。。。？
- 分治法
  - 先按x坐标排序，递归两边求解
  - 合并时：
    - 左右两侧答案取较小值a
    - 距离中轴线超过a的点是无用的
    - 按y坐标排序（可以归并）
    - 每个点只可能和前后常数个点形成有用点对
  - O(nlgn)



三维几何基础

- 平面法向量（垂直于平面的向量）（表示了平面的方向）
- 设平面上任意两不平行的向量为A、B
- 法向量F $= \begin{vmatrix} A_x & A_y & A_z \\ B_x & B_y & B_z \\ i & j & k \end{vmatrix}$
- 可以用来计算平面之间的夹角关系、点到平面的距离等
- 推导和二维类似，但需要一些想象力

三维和高维问题

- 三维问题
  - 通常不会太难
- 高维问题
  - 考虑用线性代数求解

休息一下

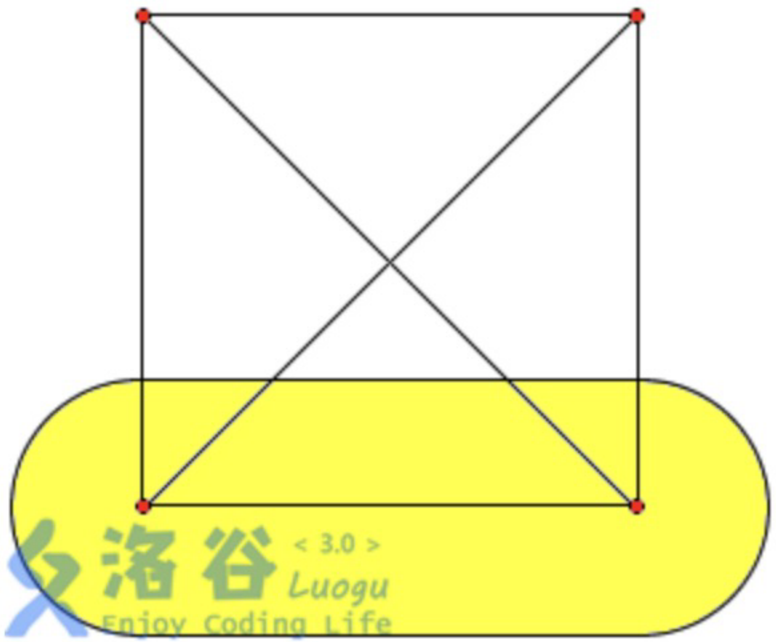
最小圆覆盖

- 给出N=100000个点，让你画一个最小的包含所有点的圆。
- 每个点作为圆的边界点的概率是3/N
- 如果圆边界有2个点，则这两个点是直径，如果有三个，则是外接圆
- 任取2个点AB作为圆的边界，依次枚举其他点，如果点C不在圆AB里，则C必然是前k个点中的边界点之一（确定了一个点）
- 再去确定第2个点、第3个点
- 期望时间复杂度O(N)
- 随机打乱输入以确保能达到期望复杂度

小 Z 是一位杰出的数学家。聪明的他特别喜欢研究一些数学小问题。

有一天，他在一张纸上选择了 n 个点，并用铅笔将它们两两连接起来，构成 $\frac{n(n-1)}{2}$ 条线段。由于铅笔很细，可以认为这些线段的宽度为 0。

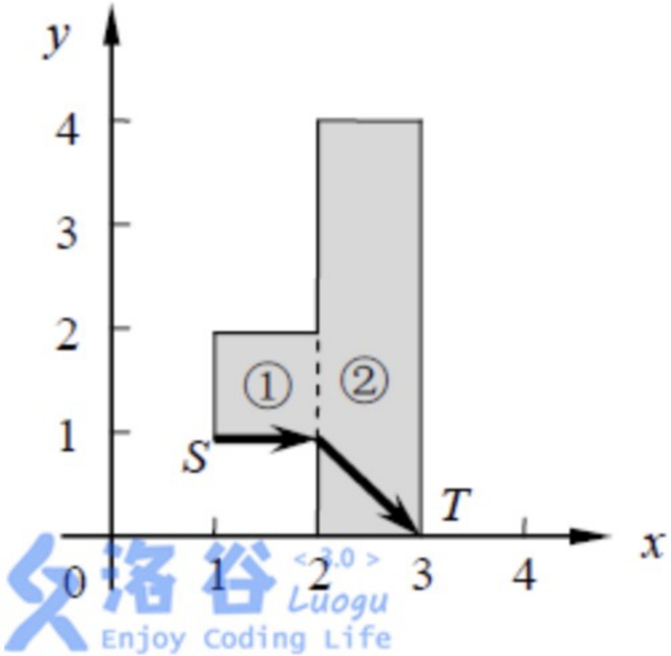
望着这些线段，小 Z 陷入了冥想中。他认为这些线段中的一部分比较重要，需要进行强调。因此小 Z 拿出了毛笔，将它们重新进行了描边。毛笔画在纸上，会形成一个半径为 r 的圆。在对一条线段进行描边时，毛笔的中心（即圆心）将从线段的一个端点开始，沿着该线段描向另一个端点。下图即为在一张 4 个点的图中，对其中一条线段进行描边强调后的情况。



现在，小 Z 非常想知道在描边之后纸面上共有多大面积的区域被强调，你能帮助他解答这个问题么？

新一届智能车大赛在 JL 大学开始啦！比赛赛道可以看作是由 n 个矩形区域拼接而成（如下图所示），每个矩形的边都平行于坐标轴，第 i 个矩形区域的左下角和右上角坐标分别为 $(x_{i,1}, y_{i,1})$ 和 $(x_{i,2}, y_{i,2})$。

题目保证：$x_{i,1} < x_{i,2} = x_{i+1,1}$，且 $y_{i,1} < y_{i,2}$，相邻两个矩形一定有重叠在一起的边（如图中虚线所示），智能车可以通过这部分穿梭于矩形区域之间。



选手们需要在最快的时间内让自己设计的智能车从一个给定的起点 S 点到达一个给定的终点 T 点，且智能车不能跑出赛道。假定智能车的速度恒为 v 且转向不消耗任何时间，你能算出最快需要多少时间完成比赛么？

水平可见直线

- 以y=kx+b的形式给出10w条直线
- 求从y轴正方向无限远处观看，能看到哪些直线（没有被遮挡）

水平可见直线

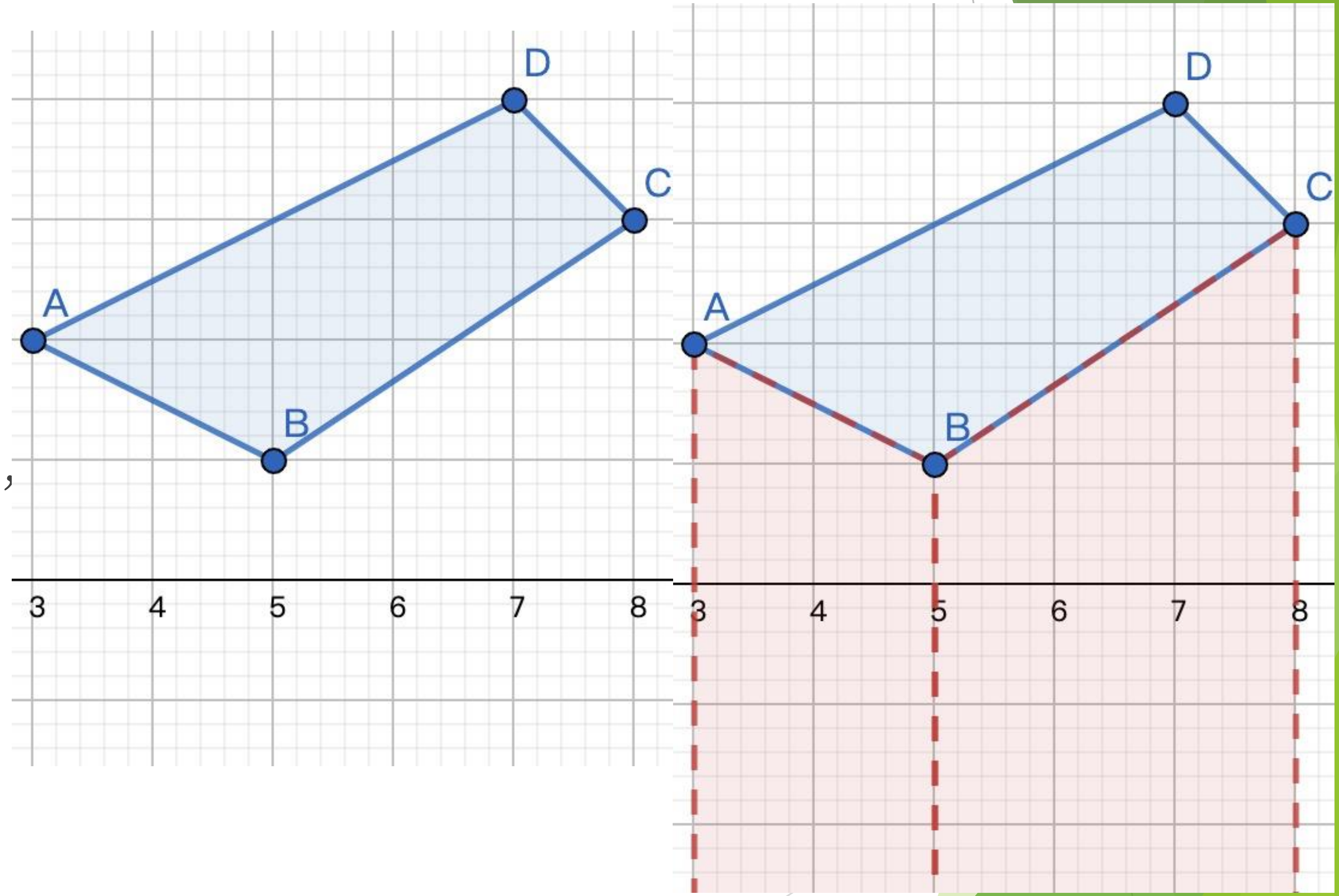
- 以$y=kx+b$的形式给出10w条直线
- 求从y轴正方向无限远处观看，能看到哪些直线（没有被遮挡）
- 类似于凸包的方案，按斜率排序
- 若$i<j<k$，直线L_i和L_j的交点为x_1，L_i和L_k的交点为x_2且$x_1>x_2$，则L_j看不见
- 维护一个栈
- 注意斜率相同的直线

冰冻青蛙

- Cirno闲着无事的时候喜欢冰冻青蛙。
  Cirno每次从雾之湖中固定的n=1000个结点中选出一些点构成一个简单多边形，Cirno运用自己的能力能将此多边形内所有青蛙冰冻。
  雾之湖生活着m=1000只青蛙,青蛙有大有小，所以每只青蛙的价值为一个不大于10000的正整数。
- 有Q=10000个询问， 每个询问给出一个多边形，问青蛙的总价值

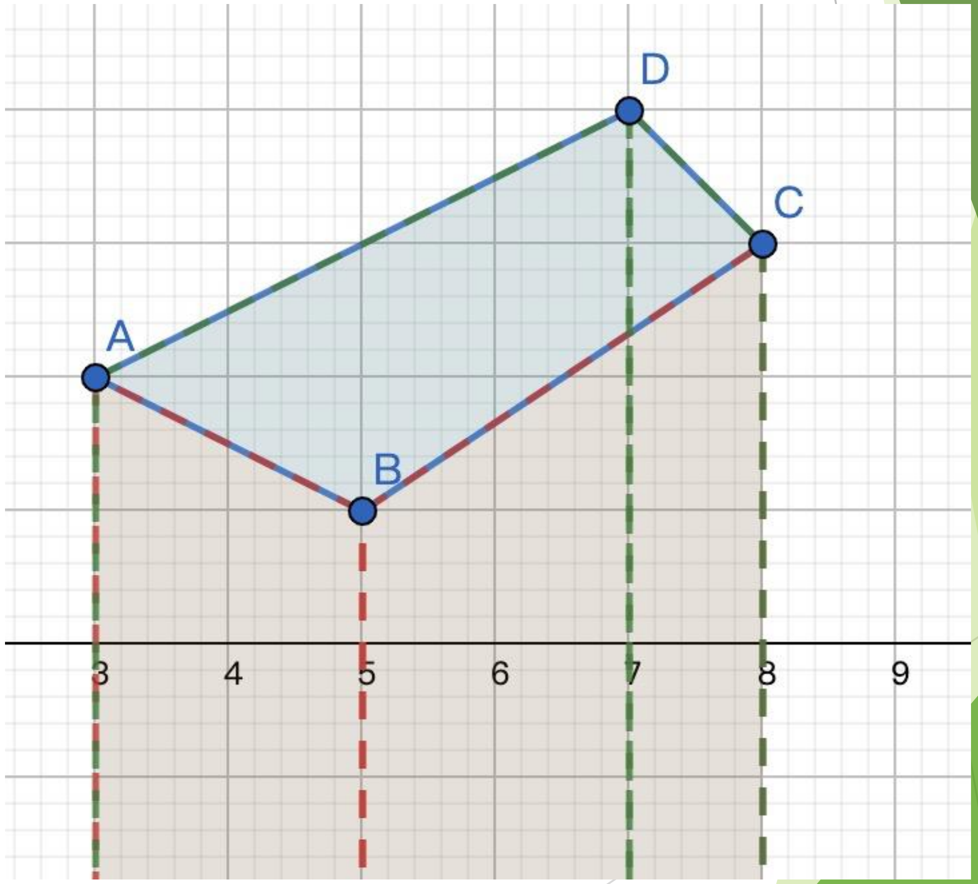
冰冻青蛙

- 考虑到只有1000个点可以作为多边形边界，可以枚举其中点对
- 对于一个多边形，可以视为一些无限长梯形的“差’
- 预处理O(n^2)
- 询问O(输入规模)



冰冻青蛙

- 考虑到只有1000个点可以作为多边形边界，可以枚举其中点对
- 对于一个多边形，可以视为一些无限长梯形的“差”
- 预处理O(n^2)
- 询问O(输入规模)



矩形面积和

- 平面上有10w个矩形，边界平行于坐标轴
- 求面积并

平面图

在一个平面中有 n 个顶点和 m 条直线段，第 i 个顶点的坐标为 (x_i, y_i)， 第j条直线段连接顶点 u_j 和顶点 v_j，权值为 h_j， 除顶点 u_j 和 v_j 外直线段 j 不经过其他的顶点。 任意两条直线段如果存在公共点，则该公共点一定是一个顶点，此时这两条直线段都会连接这个顶点。 对于任意的两个顶点 x 和 y，总是可以找到一顶点序列 $a_1, a_2, \dots, a_k$ 使得 $a_1 = x, a_k = y$ 且对于任意 $1 \le i < k$ 满足 a_i 和 a_{i+1}被一条直线段直接连接。

这 m 条直线段将整个平面分成了若干个区域，其中只有一个区域是无穷大的，其余均是有界的，我们称无穷大的区域为禁区。

现在给出 q 次询问，每次给定平面中的任意两个不是顶点且分别不在任意一条直线段上的点 A 和 B， 请画一条曲线连接 A 和 B，要求曲线不能经过禁区以及任何顶点，并使得穿过的直线段中权值最大的尽可能小。 你需要对每次询问回答这个值最小为多少。

<table>
<tr><th>编号</th><th>n、m、q 的范围</th><th>x、y 的范围</th><th>其他特征</th></tr>
<tr><td>1</td><td>$n = 10, m = 13, q = 20$</td><td rowspan="2">$0 \le x \le 1$
$0 \le y \le 2000$</td><td rowspan="5">所有直线段的长度
均等于 1</td></tr>
<tr><td>2</td><td>$n = 2012, m = 3016$
$q \le 1000$</td></tr>
<tr><td>3</td><td>$n, m \le 50, q \le 200$</td><td rowspan="3">$0 \le x, y \le 1000$</td></tr>
<tr><td>4</td><td rowspan="2">$n, m \le 100{,}000$
$q \le 100{,}000$</td></tr>
<tr><td>5</td></tr>
<tr><td>6</td><td rowspan="2">$n, m \le 2{,}000$
$q \le 2{,}000$</td><td rowspan="5">$0 \le x, y \le 10^7$</td><td>每个有界区域都是凸多边形
且每条直线段的权值均等于 1</td></tr>
<tr><td>7</td><td>每条直线段的权值均等于 1</td></tr>
<tr><td>8</td><td rowspan="3">$n, m \le 100{,}000$
$q \le 100{,}000$</td><td rowspan="3">无</td></tr>
<tr><td>9</td></tr>
<tr><td>10</td></tr>
</table>

对于全部数据，均满足 $5 \le n, m, q \le 100,000$，所有直线段的权值不会超过 10^9。所有询问坐标均为不超过 10^7 的实数，且保证是 0.5 的整数倍。

一些杂谈

- 计算几何常数
  - 通常很大
  - 浮点运算多，乘除运算多
- 精度问题
  - 遇到浮点问题时，往往需要注意精度处理
- C++
  - C++中的三角函数采用弧度制（$360^o = 2\pi$）
- 调试方法
  - 作图

一些杂谈

- 计算几何的特点是：嘴巴选手往往容易AC
- 知易行难，实践为先

谢谢大家