计算几何

$n + e$

Tsinghua University

2017 年 7 月 20 日

About

- 需要一定的平面几何与**解析几何**知识、数形结合思想
- 考验选手代码能力（其实主要就是写一个个小函数）
- 有的时候往往嘴巴 AC，测一下 WA 光了……

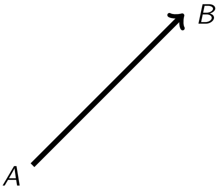
1. 必备技能
   - 向量
     - 基本定义
     - 向量运算
   - 数据存贮
   - 公式定理
   - 小应用
2. 基本算法
3. 实战演练

基本定义

- 向量 = 矢量，一个有方向、大小的量。数对应着标量。

```
struct P{double x,y;}a;    ……    a=(P){xa,ya};
```

- $A(x_A, y_A)$，$B(x_B, y_B)$，$\overrightarrow{AB} = B - A = (x_B - x_A, y_B - y_A)$。终-始



基本定义

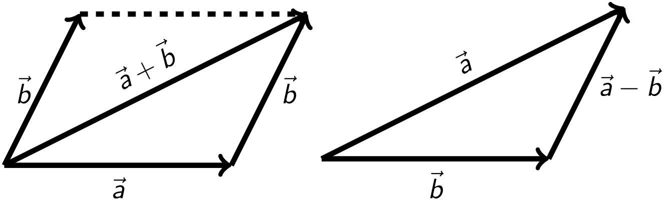
- 二维平面上，一个点 A 与向量 $\overrightarrow{OA}=\vec{a}=(x_A,y_A)$ 等价，程序实现时不对点与向量作区分。向量通常用小写字母表示
- 点的 x-y 排序：

```
bool operator<(const P&a,const P&b){
    return a.x<b.x||a.x==b.x&&a.y<b.y;
}
```

注意精度差

向量运算

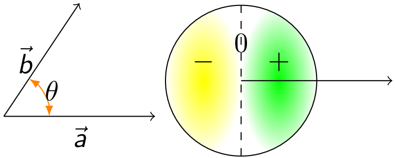
1. 加法 (平行四边形法则): $\overrightarrow{OC} = (x_A + x_B, y_A + y_B)$, OACB 构成一个平行四边形。
2. 减法: $\vec{b} - \vec{a} = \overrightarrow{AB} = (x_B - x_A, y_B - y_A)$
3. 乘/除一个数 p: 对应着扩大/缩小 p 倍 $\overrightarrow{OA} \cdot p = (x_A \cdot p, y_A \cdot p)$
4. 模长: $|\overrightarrow{OA}| = \sqrt{x_A^2 + y_A^2}$



5. 点积：$\vec{a}\cdot\vec{b}=\vec{b}\cdot\vec{a}=x_Ax_B+y_Ay_B=|\vec{a}||\vec{b}|cos<\vec{a},\vec{b}>$

利用点积求夹角：acos(dot(a,b)/len(a)/len(b))

$\vec{a}\cdot\vec{b}=0$ 等价于 $\vec{a}\perp\vec{b}$

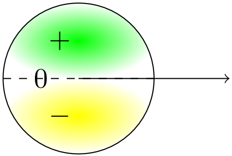


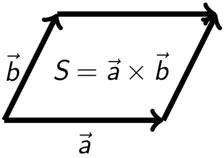
注意 C++ 中的角度均采用弧度制，$\pi=180°,\frac{\pi}{2}=90°$

不要手打 π：const double pi=acos(-1);

手打会出事，有较大精度差：写个 *FFT* 就懂了。

6. 叉积：$\vec{a} \times \vec{b} = -\vec{b} \times \vec{a} = x_A y_B - x_B y_A$
$|\vec{a} \times \vec{b}| = |\vec{a}||\vec{b}| sin < \vec{a}, \vec{b} >$
$\vec{a} \times \vec{b} = 0$ 等价于 $\vec{a}, \vec{b}$ 共线（可以反向）
$\vec{a} \times \vec{b} > 0$：$\vec{b}$ 在 $\vec{a}$ 左侧
$\vec{a} \times \vec{b} < 0$：$\vec{b}$ 在 $\vec{a}$ 右侧
叉积求面积：有向面积，如下图





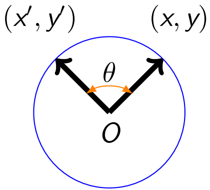
7. 旋转：向量 (x, y) 绕坐标原点逆时针旋转 θ 度

$$x' = x\cos\theta - y\sin\theta$$

$$y' = x\sin\theta + y\cos\theta$$

有兴趣了解具体原因的同学，课间可以找我了解复平面的那套理论：

$$(x + yi)(\cos\theta + i\sin\theta) = x\cos\theta - y\sin\theta + (x\sin\theta + y\cos\theta)i$$



一个很常用的小技巧

- 在读入的时候将坐标系上的每个点进行微小扰动

```
P a=(P){x+eps(),y+eps()};
P a=rotate(a,1e-7);
```

- 避免出现斜率不存在的情况
- 误差？本来就有

1. 必备技能
   - 向量
   - 数据存贮
   - 公式定理
   - 小应用
2. 基本算法
3. 实战演练

1. 直线/线段：解析式/两点坐标

$$Ax + By + C = 0 \text{ or } y = kx + b$$

2. 圆：圆心坐标，圆的半径

$$(x-a)^2 + (y-b)^2 = r^2$$

3. 多边形：多边形上所有点坐标（推荐使用逆时针）
4. 半平面：

$$Ax + By + C \geq 0$$

1. 必备技能
   - 向量
   - 数据存贮
   - 公式定理
   - 小应用
2. 基本算法
3. 实战演练

1. 正弦定理：

$$\frac{a}{\sin A} = \frac{b}{\sin B} = \frac{c}{\sin C} = 2R$$

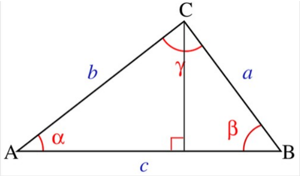
其中 R 是三角形外接圆半径

应用：知 AAS/ASA 解三角形

2. 余弦定理：

$$a^2 = b^2 + c^2 - 2bc\cos A$$
$$b^2 = a^2 + c^2 - 2ac\cos B$$
$$c^2 = a^2 + b^2 - 2ab\cos C$$

应用：知 SAS/SSS 解三角形



3. 质心：点集的加权平均数，就是好几个杠杆原理

$$x_0 = \frac{1}{N}\sum_{i=1}^{N} x_i m_i,\ y_0 = \frac{1}{N}\sum_{i=1}^{N} y_i m_i$$

三角形质心：中线交点

四边形质心：两个三角形质心的质心，权重为三角形面积

定比分点：杠杆原理坐标化

4. 求三角形面积的几种姿势
   - 叉积/2
   - $S = \frac{1}{2}ab\sin C = \frac{1}{2}bc\sin A = \frac{1}{2}ac\sin B$
   - 海伦公式：$p = (a+b+c)/2,\ S = \sqrt{p(p-a)(p-b)(p-c)}$

1. **必备技能**
   - 向量
   - 数据存贮
   - 公式定理
   - 小应用
2. 基本算法
3. 实战演练

1. 折线段的拐向判断 P_0, P_1, P_2

1. 折线段的拐向判断 P_0, P_1, P_2 $(P_1 - P_0) \times (P_2 - P_0)$
2. 判断点是否在线段上 P,A,B

1. 折线段的拐向判断 P_0, P_1, P_2 $(P_1 - P_0) \times (P_2 - P_0)$
2. 判断点是否在线段上 P,A,B $|PA| + |PB| = |AB|$
3. 判断两线段是否规范相交 (A,B),(C,D)

   规范相交：两线段恰好有一个公共点，且不在任何一条线段的端点。等价于每条线段的两个端点都在另一条线段的两侧。

1. 折线段的拐向判断 P_0, P_1, P_2 $(P_1 - P_0) \times (P_2 - P_0)$
2. 判断点是否在线段上 P,A,B $|PA| + |PB| = |AB|$
3. 判断两线段是否规范相交 (A,B),(C,D)

规范相交：两线段恰好有一个公共点，且不在任何一条线段的端点。等价于每条线段的两个端点都在另一条线段的两侧。

$$((C - A) \times (D - A)) \cdot ((C - B) \times (D - B)) < 0$$

如果 A、B 看 CD 两点的相对位置是反过来的，那么 A、B 在 CD 两侧

同理有

$$((A - C) \times (B - C)) \cdot ((A - D) \times (B - D)) < 0$$

两个式子同时成立，则称线段 AB 与线段 CD **规范相交**

如果把小于号改成小等号？有一些奇怪的情况会发生

4. 判断线段和直线是否相交

4. 判断线段和直线是否相交 可以像刚才那样做，不过可以这样：求出 $kx-y+b=0$ 或者 $Ax+By+C=0$，强行带点进去看看左边算出来什么结果，如果两个值异号则相交
5. 判断矩形是否与直线相交

4. 判断线段和直线是否相交 可以像刚才那样做，不过可以这样：求出 $kx - y + b = 0$ 或者 $Ax + By + C = 0$，强行带点进去看看左边算出来什么结果，如果两个值异号则相交
5. 判断矩形是否与直线相交 和某条边有交就行，不过可以玩玩对角线
6. 判断点是否在多边形中

4. 判断线段和直线是否相交 可以像刚才那样做，不过可以这样：求出 $kx - y + b = 0$ 或者 $Ax + By + C = 0$，强行带点进去看看左边算出来什么结果，如果两个值异号则相交
5. 判断矩形是否与直线相交 和某条边有交就行，不过可以玩玩对角线
6. 判断点是否在多边形中 射线法
7. 判断线段是否在多边形内

4. 判断线段和直线是否相交 可以像刚才那样做，不过可以这样：求出 $kx-y+b=0$ 或者 $Ax+By+C=0$，强行带点进去看看左边算出来什么结果，如果两个值异号则相交
5. 判断矩形是否与直线相交 和某条边有交就行，不过可以玩玩对角线
6. 判断点是否在多边形中 射线法
7. 判断线段是否在多边形内 求交点，排序，取相邻中点 check
   两个端点都在多边形内部并不能说明线段在多边形内部：凹
8. 计算点到线段的距离 $P(x_0, y_0)$

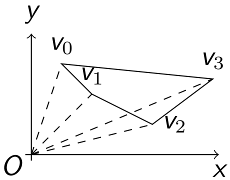
4. 判断线段和直线是否相交 可以像刚才那样做，不过可以这样：求出 $kx-y+b=0$ 或者 $Ax+By+C=0$，强行带点进去看看左边算出来什么结果，如果两个值异号则相交
5. 判断矩形是否与直线相交 和某条边有交就行，不过可以玩玩对角线
6. 判断点是否在多边形中 射线法
7. 判断线段是否在多边形内 求交点，排序，取相邻中点 check
   两个端点都在多边形内部并不能说明线段在多边形内部：凹
8. 计算点到线段的距离 $P(x_0,y_0)$ 先看端点，再求距离

$$l: Ax+By+C=0 \ \rightarrow \ d=\frac{|Ax_0+By_0+C|}{\sqrt{A^2+B^2}}$$

9. 计算 N 边形面积

9. 计算 N 边形面积

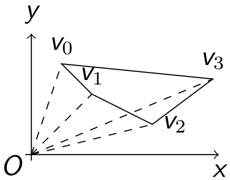
$$S = \frac{1}{2}\left|\sum_{i=0}^{N-1}(P_i \times P_{(i+1) \bmod N})\right|$$



10. 计算线段或直线与线段的交点

9. 计算 N 边形面积

$$S = \frac{1}{2}\left|\sum_{i=0}^{N-1}(P_i \times P_{(i+1) \bmod N})\right|$$

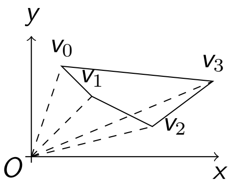


10. 计算线段或直线与线段的交点

先变成直线，暴力解方程，然后检验这个点符不符合条件。

9. 计算 N 边形面积

$$S = \frac{1}{2}\left|\sum_{i=0}^{N-1}(P_i \times P_{(i+1) \bmod N})\right|$$



10. 计算线段或直线与线段的交点
    先变成直线，暴力解方程，然后检验这个点符不符合条件。
    → 千万不要去用向量做：有这么简洁明了的方法不用，去用那个拼命分类讨论又臭又长还会 WA 的方法，简直是作死！

11. 求过不共线三点的圆

11. 求过不共线三点的圆 暴力解方程
12. 求线段或直线与圆的交点、求圆与圆的交点（假设相交）

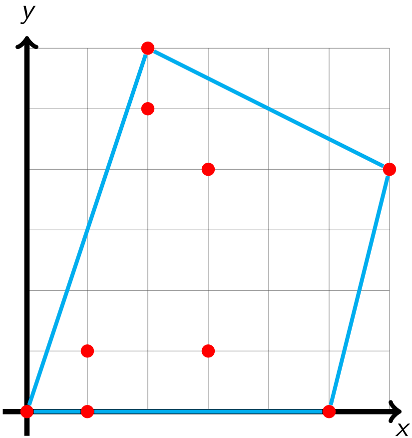
11. 求过不共线三点的圆 暴力解方程
12. 求线段或直线与圆的交点、求圆与圆的交点（假设相交）
    解一元二次方程 or 余弦定理
13. 求过圆外某点的两条圆的切线

11. 求过不共线三点的圆 暴力解方程
12. 求线段或直线与圆的交点、求圆与圆的交点（假设相交）
    解一元二次方程 or 余弦定理
13. 求过圆外某点的两条圆的切线 强行解析几何 $\triangle = 0$？
14. 求两圆的外公切线

11. 求过不共线三点的圆 暴力解方程
12. 求线段或直线与圆的交点、求圆与圆的交点（假设相交）
    解一元二次方程 or 余弦定理
13. 求过圆外某点的两条圆的切线 强行解析几何 $\triangle = 0$?
14. 求两圆的外公切线 相似三角形解方程

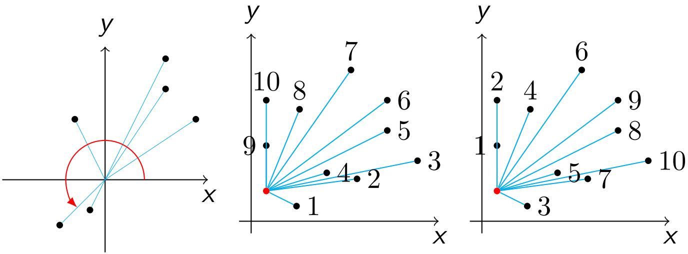
- 解几大法好!
- 解方程大法好!
- 当你毫无头绪的时候，就想想怎么暴力解方程就行了哈哈哈哈

什么是凸包



排序

- x-y 排序：std::sort(a+1,a+1+n); 小于号最前面定义过了
- 极角排序：用一用 atan2(y,x)

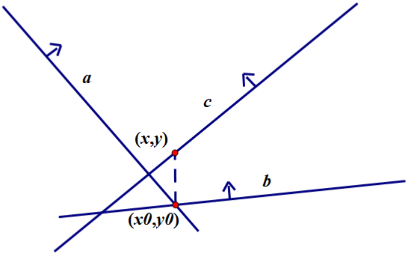


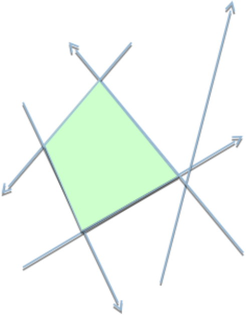
- 个人推荐使用 x-y 排序，极角排序效率不高并且有精度差，特定条件下才使用。

如何求凸包

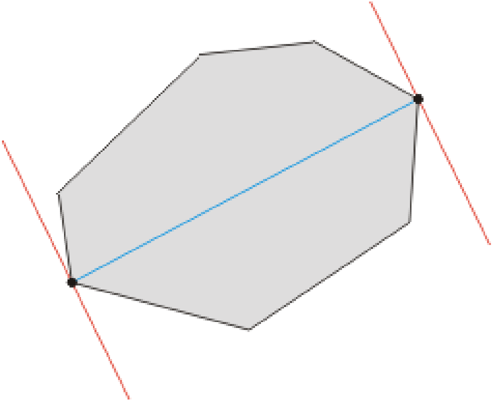
```cpp
std::sort(a+1,a+1+n);
#define calc(a,b,c) ((b-a)*(c-a))
for(int i=1;i<=n;i++){//下凸壳
    while(1<t&&calc(a[q[t-1]],a[q[t]],a[i])<=0)t--;
    q[++t]=i;
}
int tmp=t;
for(int i=n-1;i;i--){//上凸壳
    while(tmp<t&&calc(a[q[t-1]],a[q[t]],a[i])<=0)t--;
    q[++t]=i;
}
```

- 求形如 n 条 $y \geq kx + b$ 的半平面的并，答案一定是形如凸包的下凸壳
- 这样对偶：将一条直线的 (k, b) 视为一个点 $(k, -b)$，然后做凸包的下凸壳
- 我专门写了一篇Blog讲原理





- 用一对平行且与凸包相切的直线在凸包上进行扫描。
- 用两个指针维护，~~叉积比较所指下一条边的极角~~。根本不用比角度，用三角形面积就好了。面积最大的时候停下来。



1 必备技能

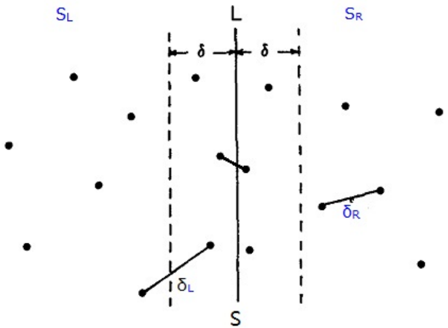
2 基本算法

- 凸包
- 半平面交
- 旋转卡壳
- 最近点对
- 最小圆覆盖
- 自适应辛普森积分

3 实战演练

- 平面上有 N 个点，求欧几里德距离最近的点对。$N \leq 100000$

- 平面上有 N 个点，求欧几里德距离最近的点对。$N \le 100000$
- 分治法求解步骤：$O(NlogN)$
  1. 将点集 S 分为两个子集 S_L 和 S_R 分别求解
  2. 记 δ 为子集中求得的最优值（$\min(\delta_L, \delta_R)$），合并两个集合求解。图中以分界线为中心，任何一个 $2\delta \cdot 2\delta$ 的正方形内，只有常数个点，暴力 for 过去就好了。



人类智慧

- 随机转坐标系，每 20 个点一组，块内暴力，块外不管
- 会 WA？多转几次！
- 调完参跑得比谁都快

1 必备技能

2 基本算法

- 凸包
- 半平面交
- 旋转卡壳
- 最近点对
- **最小圆覆盖**
- 自适应辛普森积分

3 实战演练

- 平面上有 n 个点，求一个半径最小的圆覆盖所有点。
- 就是暴力枚举三个点，构造一个圆，如果接下来的点不在圆内的话，那么以这个点为圆的端点之一，再来暴力 for
- 根据概率与期望的那套理论，表面上是 $O(N^3)$，实际上效率是 $O(N)$ 的。只要出题人和你没有杀父之仇就不会被卡。

```
std::random_shuffle(a+1,a+1+n);
for(make_circle(a[1],a[2]),i=3;i<n;i++)if(dis(a[i],o)>r)
for(make_circle(a[1],a[i]),j=2;j<i;j++)if(dis(a[j],o)>r)
for(make_circle(a[j],a[i]),k=1;k<j;k++)if(dis(a[k],o)>r)
    make_circle(a[i],a[j],a[k]);
```

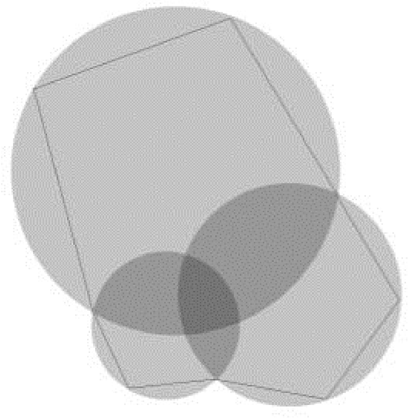
1 必备技能

2 基本算法

- 凸包
- 半平面交
- 旋转卡壳
- 最近点对
- 最小圆覆盖
- 自适应辛普森积分

3 实战演练

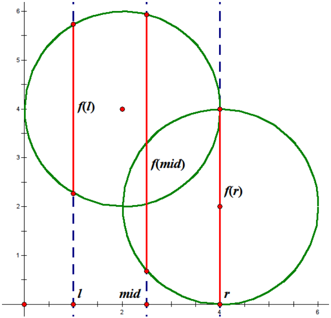
- 求若干圆和圆/多边形的面积并



- 正规解法：扫描线……好难写……

$$F(l,r)=\int_l^r f(x)\,dx \approx \frac{r-l}{6}\left(f(l)+4f\left(\frac{l+r}{2}\right)+f(r)\right)$$

- f(x) 若是三次及三次以下的函数，则直接取等
- 对于一般的函数：暴力递归，至两边误差小于 eps



对于区间 $[l, r]$，如果在这段区间中将 $f(x)$ 当做一个二次函数，那么很容易得到如下的近似：

$$F(l, r) = \int_l^r f(x)\mathrm{d}x = \frac{r-l}{6}\Big(f(l) + f(r) + 4f(mid)\Big), mid = \frac{l+r}{2}$$

用这个公式计算区间 $[l, r], [l, mid], [mid, r]$ 的值，若

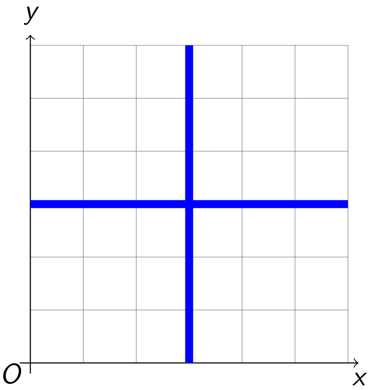
$$|F(l, r) - F(l, mid) - F(mid, r)| > \epsilon$$

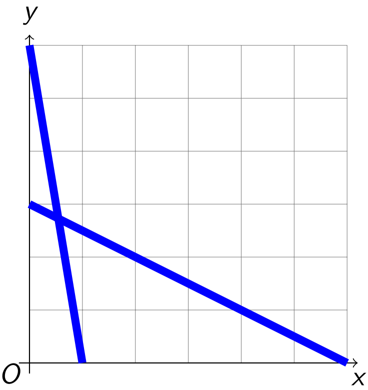
那么递归计算 $[l, mid], [mid, r]$，否则可以认为这次模拟足够精确，直接用模拟值作为返回值即可。

ϵ 越小，程序运行时间越长，得到的结果就越精确。如果 ϵ 设得合理的话，那么这个算法是可以通过此题的。

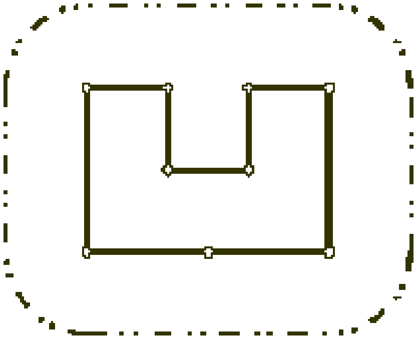
- 两条线段组成一个固定图形，不能旋转，能容纳多少面积的雨水。根据生活常识，雨是从天上垂直下落的。

- 两条线段组成一个固定图形，不能旋转，能容纳多少面积的雨水。根据生活常识，雨是从天上垂直下落的。
- 如果不交，则接不了
- 如果交了，就一定能接水吗？

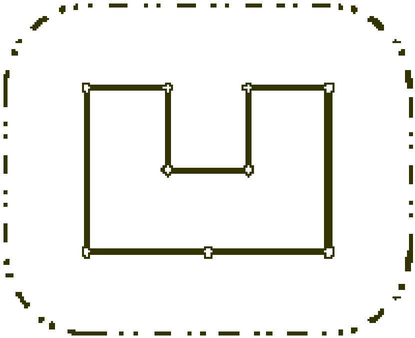




- 给出平面上若干个点的坐标，让建一个环形围墙，把所有的点围在里面，且围墙距所有点的距离不小于 l。求围墙的最小长度。



- 给出平面上若干个点的坐标，让建一个环形围墙，把所有的点围在里面，且围墙距所有点的距离不小于 l。求围墙的最小长度。



- 凸包周长 $+2\pi l$

- 在某块平面土地上有 N 个点，你可以选择其中的任意四个点，将这片土地围起来，当然，你希望这四个点围成的多边形面积最大。$N \leq 2000$

- 在某块平面土地上有 N 个点，你可以选择其中的任意四个点，将这片土地围起来，当然，你希望这四个点围成的多边形面积最大。$N \leq 2000$
- 暴力枚举对角线，旋转卡壳

- 给定 n 条线段，确定是否存在一条直线，使得这 n 条线段在这条直线上的射影具有公共点。$n \leq 100$

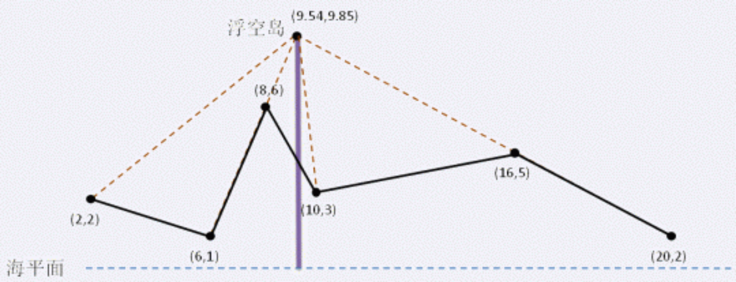
- 给定 n 条线段，确定是否存在一条直线，使得这 n 条线段在这条直线上的射影具有公共点。$n \leq 100$
- 问题转换成是否存在一根直线可以穿过所有的线段
- 任取 2 个线段端点，枚举这根直线是否和所有线段有交点，复杂度 $O(n^3)$

1 必备技能

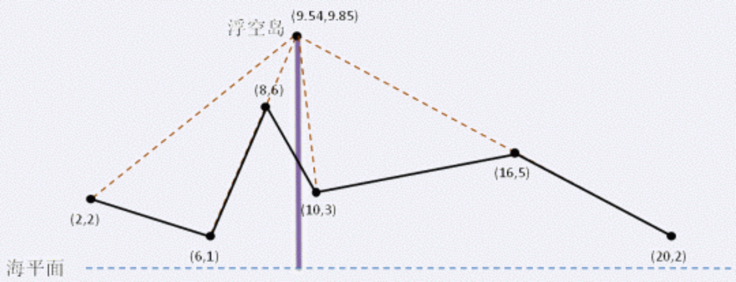
2 基本算法

3 实战演练

- 给定一座山脉，用二维坐标表示为 (x,h)，h 表示海拔高度，求能够看到山脉全貌的最低海拔高度。$N \le 1000000$



- 给定一座山脉，用二维坐标表示为 (x,h)，h 表示海拔高度，求能够看到山脉全貌的最低海拔高度。$N \le 1000000$



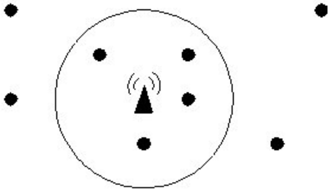
- 相邻两个点连线组成一个半平面，直接半平面交就好了。

- 给定一个凸多边形，求多边形中离边界最远的点到边界的距离。$N \leq 100, 0 \leq x_i, y_i \leq 10000$, 保留 5 位小数。

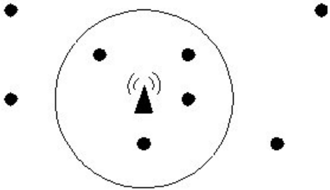
- 给定一个凸多边形，求多边形中离边界最远的点到边界的距离。$N \leq 100, 0 \leq x_i, y_i \leq 10000$, 保留 5 位小数。
- 每次将凸多边形每条边往里平移 d，判断是否存在核。二分 d 之后判断半平面交是否为空即可。空 $\rightarrow$ 缩的过头了，比标准答案大；非空 $\rightarrow$ 还能继续缩，比标准答案小。
- $N \leq 200000, |x_i|, |y_i| \leq 10^{11}$，答案精确到小数点后六位
- 不允许二分，答案精度不够，有可能超时，只能通过部分数据

- 随着 d 的增加，凸多边形变小的过程，实质是一条边一条边地减少。每过一段时间，就减少了一条边的约束，减少的那条边对凸多边形的控制可以由其相邻的两条边取代，直到最后剩下两条边。
- 那么，我们只要知道在什么时刻，减少了哪条边就可以了。
- 因此，对于每一条边 i，我们维护其在什么时刻被相邻的两条边取代，记为 t_i。
    1. 这条边相邻两边平行，则 t_i 为两边距离的一半
    2. 这条边相邻两边不平行，则 t_i 为其相邻两角的角平分线交点到它的距离
- 那么，每次对于该凸多边形，在所有边中取最小的 t_i，将所对应的边删除，并维护相邻的两条边被删除的时间就可以了。实现使用堆 + 双向链表即可。几乎没有精度差

- 在平面上放一个半径为 r 的圆使得覆盖尽量多的点。$n \leq 2000$



- 在平面上放一个半径为 r 的圆使得覆盖尽量多的点。$n \leq 2000$



- 对最优的圆进行平移，可以使圆周上至少有 1 个点。
- 以每个点为圆心，作半径为 r 的圆。若在该圆圆周上作半径为 r 的圆，要覆盖其他点，则对应一段连续的弧。（两圆求交点）
- 求覆盖次数最多的子区间。