计算几何基础

2009.02.08

- 计算几何是计算机理论科学的一个重要分支。自20世纪70年代末从算法设计与分析中独立出来起，不到30年，该学科已经有了巨大的发展，不仅产生了一系列重要的理论成果，也在众多实际领域中得到了广泛的应用。

- 计算几何题的特点
  - 代码量大
  - 特殊情况多
  - 精度问题难以控制
  - ……

需要注意的细节

- 常用头文件#include<cmath>
- 计算几何中一般来说使用double型比较频繁，请注意数据类型的选择，该用实数的时候就用double，而float容易失去精度。
- 判断double型的x是否为0，应当用fabs(x)<eps，其中eps代表某个精度，常常取eps=1e-6等。符号函数。
  还有其他类似情况也要注意精度问题，如下取整int(x+eps)，避免x=4.999999999

需要注意的细节

- 圆周率取const double PI = acos(-1.0);
- 角度制和弧度制的转换，C/C++中的三角函数均为弧度制
- 考虑到时间效率和精度，应尽量少用除法、开方、三角函数；尽量只用加、减、乘，尤其避免除以一很小的数
- 输出的时候要小心-0.00000，比如：a=-0.0000001，printf(“%.5lf”,a);

向量及其运算

- 重要工具——向量
- **定义**　　既有大小又有方向的量叫做向量。
- **表示**　　用有向线段表示和用坐标表示。
- **运算**
  - 加法、减法、数乘
  - 点积（内积）

    $\alpha \cdot \beta = x1 \times x2 + y1 \times y2 = |\alpha| \cdot |\beta| \cdot \cos\theta$
  - 叉积（外积）

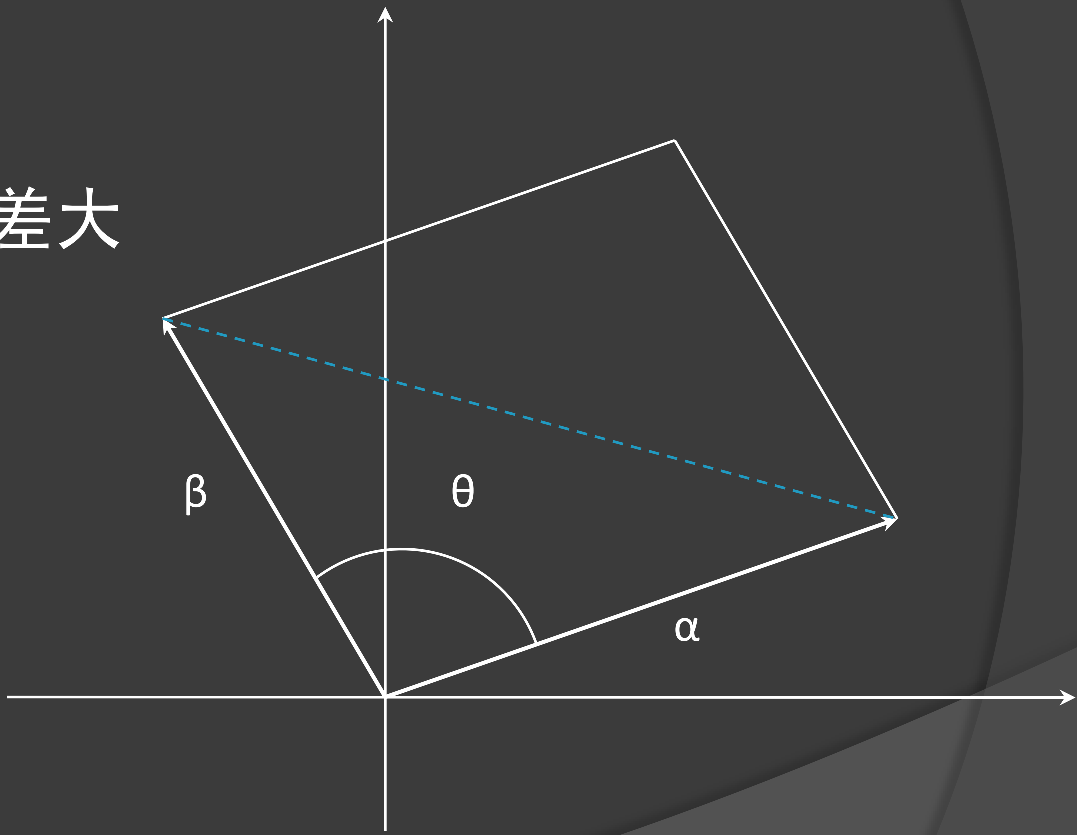
    $\alpha \times \beta = x1 \times y2 - x2 \times y1 = |\alpha| \cdot |\beta| \cdot \sin\theta$

向量乘积应用

- 求夹角
- 求面积
- 判断位置和方向

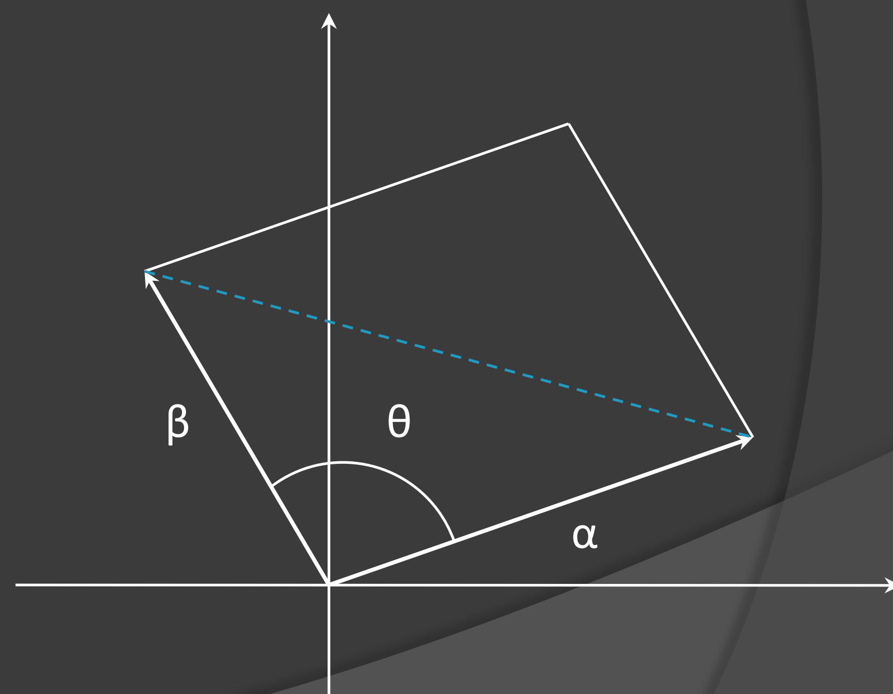
判断两向量的方向

- 解析几何：
  - 比较斜率
  - 除法代价高、误差大
  - 分母不为0



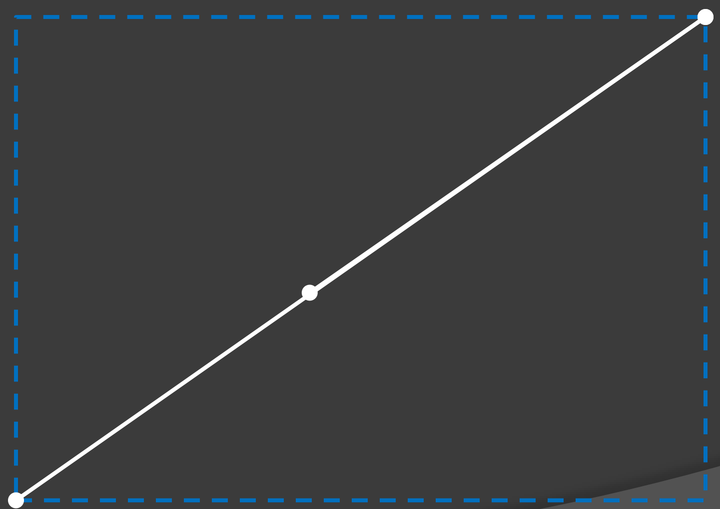
判断两向量的方向

- 判断外积的符号，右手定则
- α×β<0，β×α>0 左手系，顺时针
- α×β>0，β×α<0 右手系，逆时针
- 向左转还是向右转



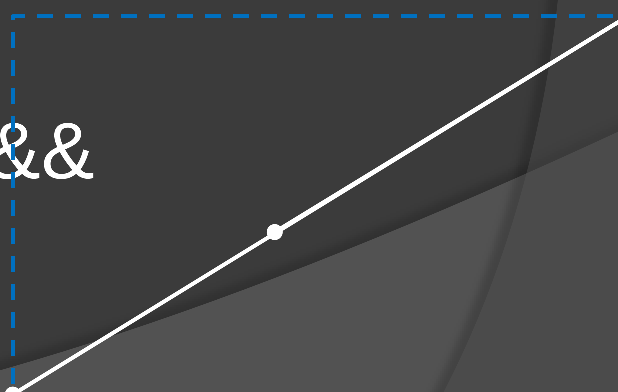
判断点在直线上

- 利用三点共线的等价条件α×β== 0
- 直线上取两不同点P1,P2，若点P在直线上，则fabs((P1 - P) × (P2 - P)) < eps



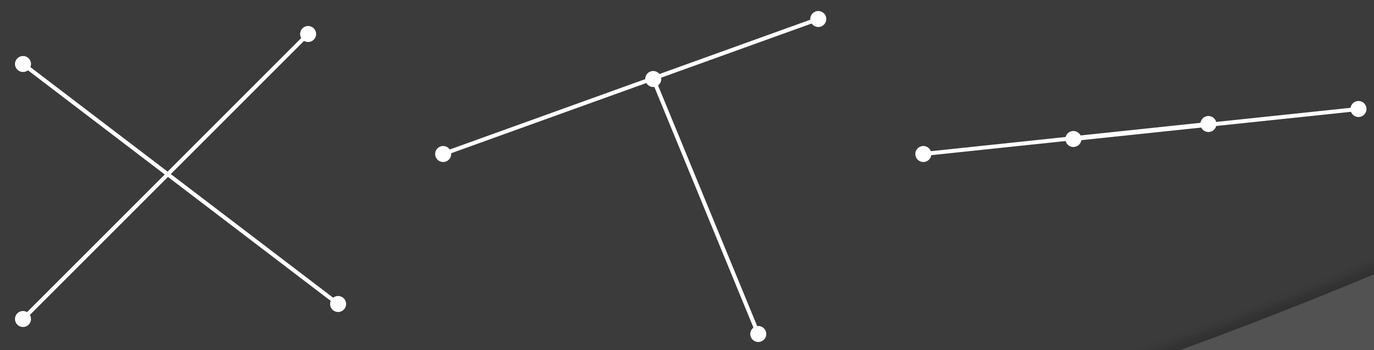
判断点在线段上

- 判断点P(x,y)是否在线段P1P2上，其中P1(x1,y1),P2(x2,y2)
- 需要验证两条
- （1）点P在P1P2所在直线上，即三点共线
- （2）点P在P1P2为对角线的矩形内
- 其中（2）利用 min(x1,x2)<=x<=max(x1,x2) && min(y1,y2)<=y<=max(y1,y2)



判断两线段相交

- 两条线段恰有惟一一个不是端点的公共点，称之为“规范相交”。
- 非规范相交：如果交点是其中一个线段的端点或者说两个线段部分重合。



判断两线段规范相交

- 两条线段相交一个必要条件：
  以两条线段为对角线的两个矩形相交。
- 用该条件加速判断
- （1）快速排斥试验：
  以P1P2为对角线的矩形S1是否和以P3P4为对角线的矩形S2相交，即
  min(x1,x2)<=max(x3,x4) &&
  min(x3,x4)<=max(x1,x2) &&
  min(y1,y2)<=max(y3,y4) &&
  min(y3,y4)<=max(y1,y2)

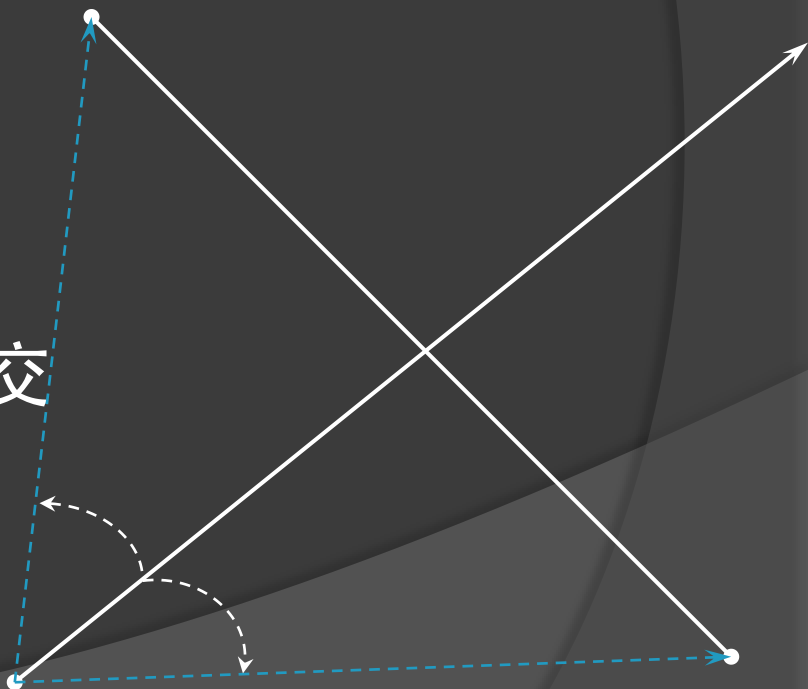
判断两线段规范相交

- （2）跨立试验：

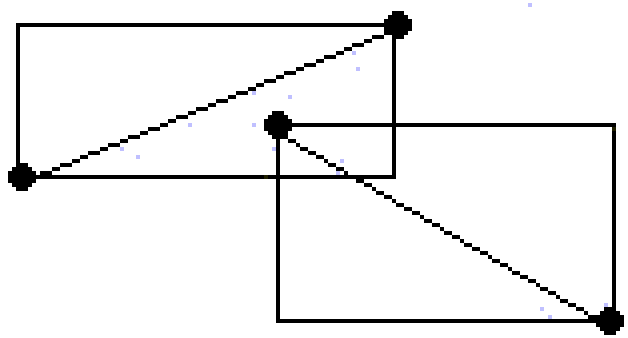
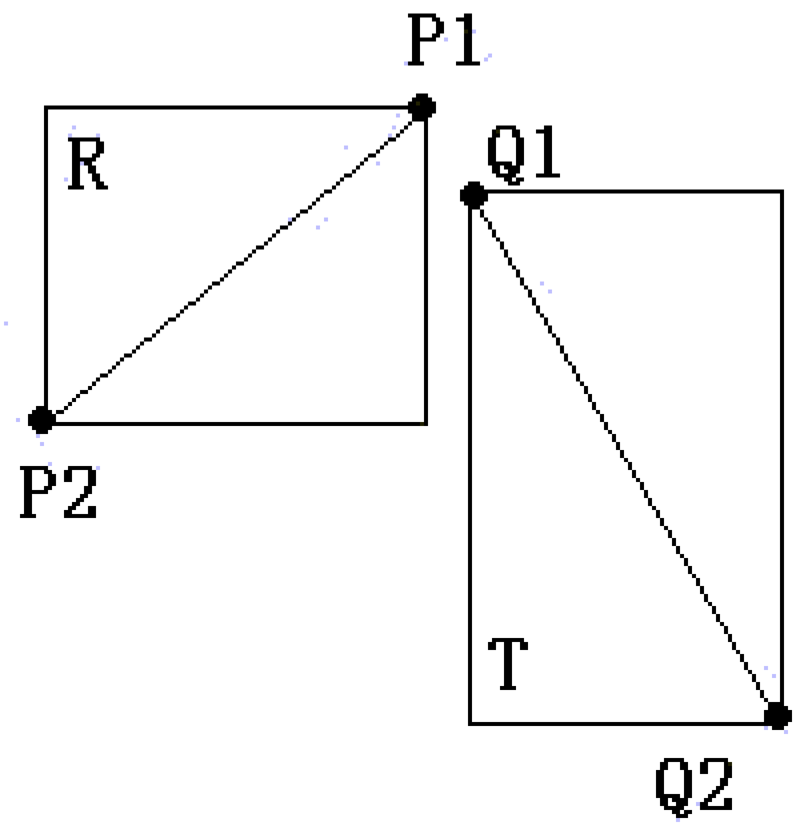
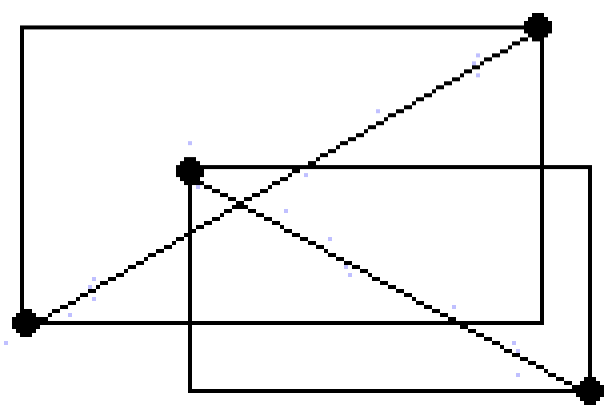
  两条线段规范相交时，每条线段两个端点都在另一条线段的异侧。

  跨立可由两叉乘异号得出

- 即为两次判断线段与直线相交



判断两线段规范相交

<table>
<tr><th></th><th>通过快速排斥实验</th><th>未通过快速排斥实验</th></tr>
<tr><td>未通过跨立实验</td><td></td><td rowspan="2">
</td></tr>
<tr><td>通过跨立实验</td><td></td></tr>
</table>

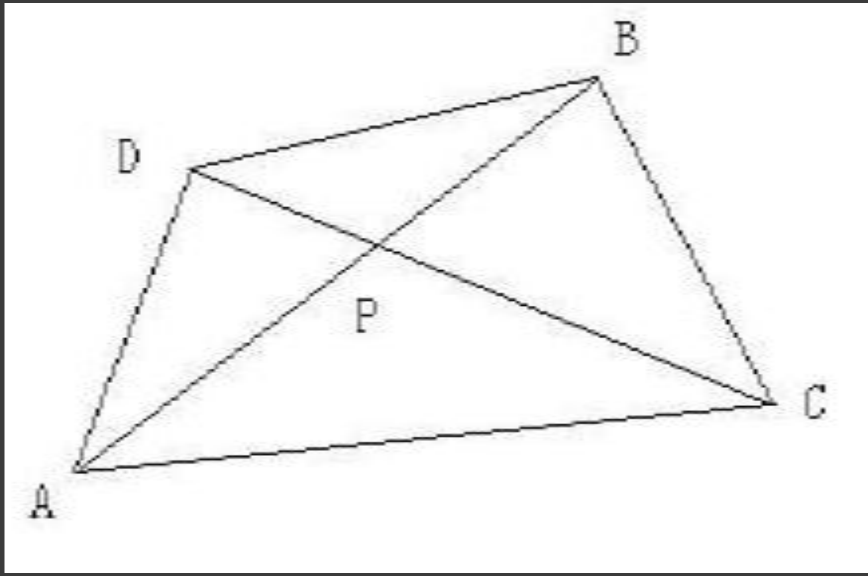
判断两线段规范相交

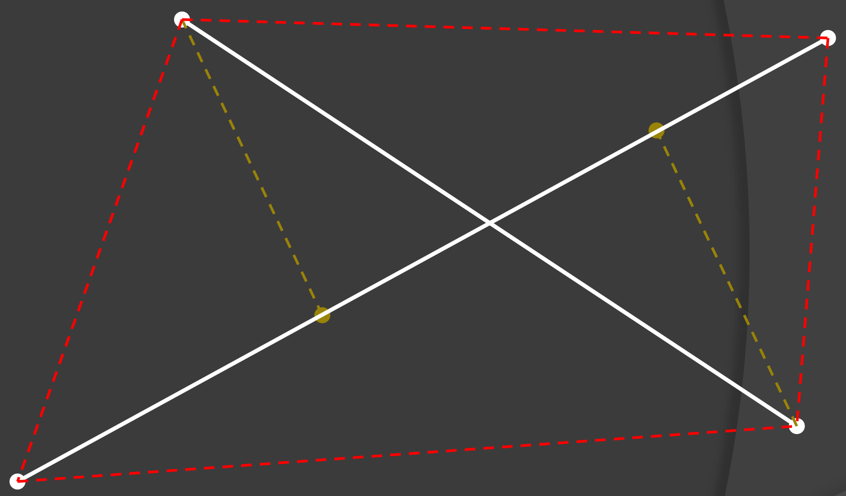
```
inline double Direction(Point a,Point b,Point p)
{
    return (b.x-a.x)*(p.y-a.y)-(b.y-a.y)*(p.x-a.x);
}

inline bool isIntersected(Point s1,Point e1,Point s2,Point e2)
{
    return max(s1.x,e1.x)>=min(s2.x,e2.x) &&
           max(s2.x,e2.x)>=min(s1.x,e1.x) &&
           max(s1.y,e1.y)>=min(s2.y,e2.y) &&
           max(s2.y,e2.y)>=min(s1.y,e1.y) &&
           Direction(s2,e2,s1)*Direction(s2,e2,e1)<-eps &&
           Direction(s1,e1,s2)*Direction(s1,e1,e2)<-eps;
}
```

练习：TOJ 1770

求两线段交点

- 利用叉积、定比分点求线段交点

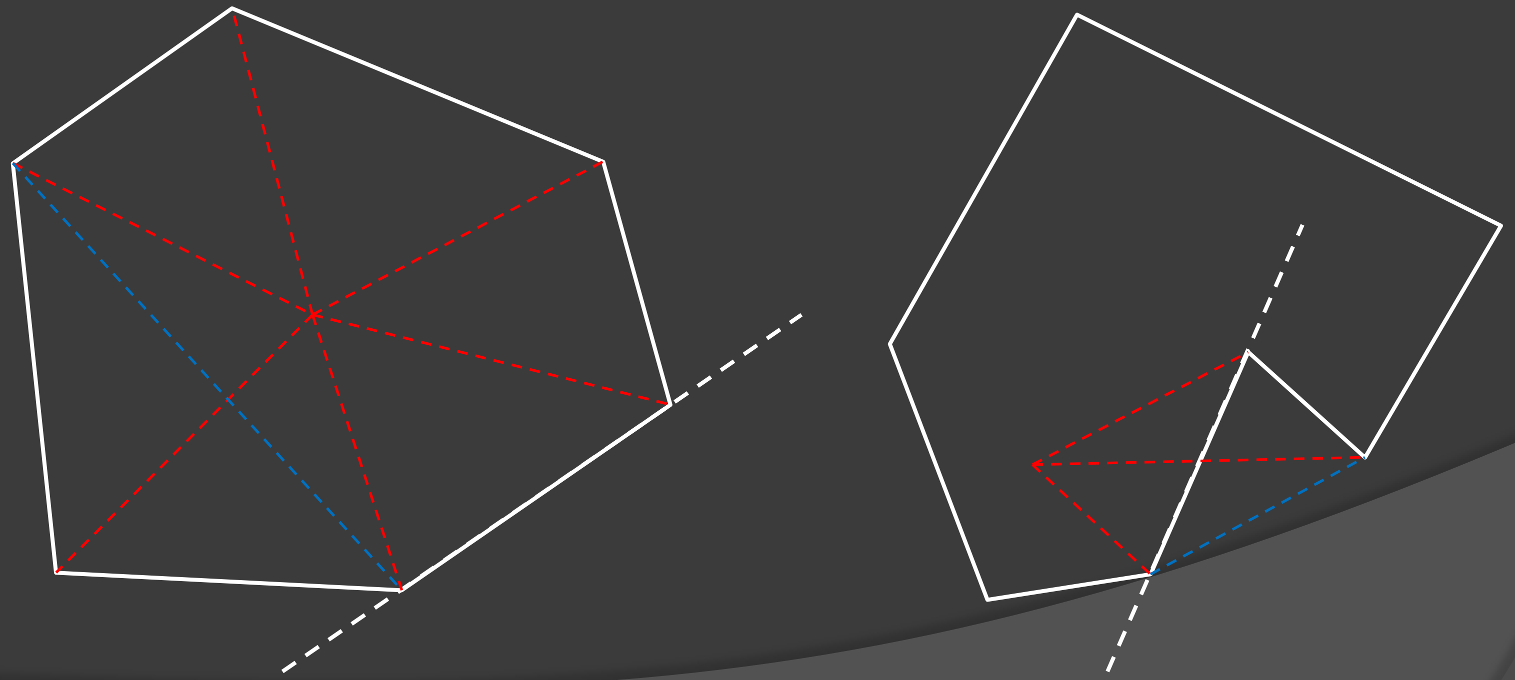




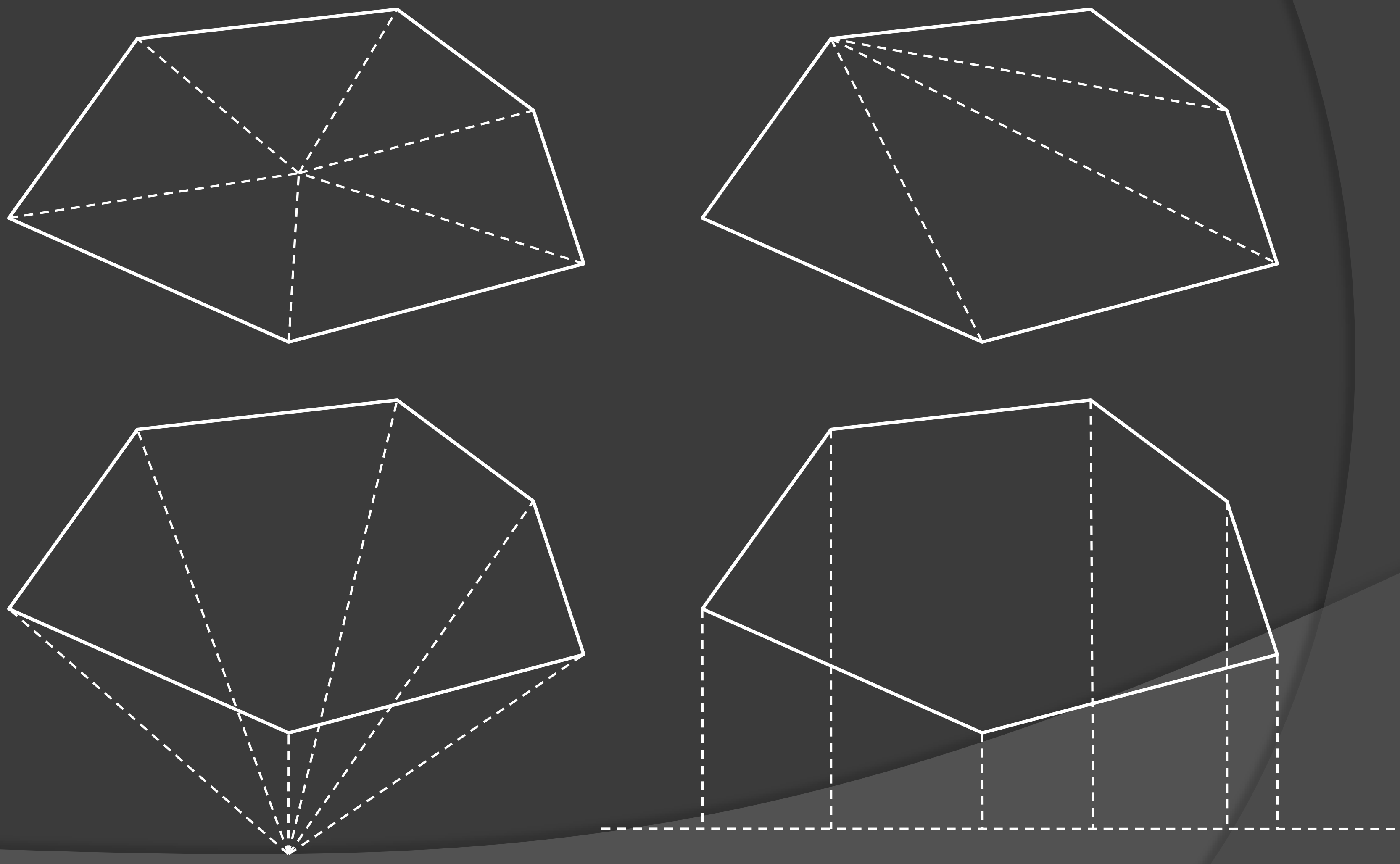
$$x_p = \frac{S_{\Delta ABD} \cdot x_C + S_{\Delta ABC} \cdot x_D}{S_{\Delta ABD} + S_{\Delta ABC}}$$

求凸多边形面积

- 过多边形任意一边做一条直线，如果其他各顶点都在这条直线的同侧，则把这个多边形叫做凸多边形

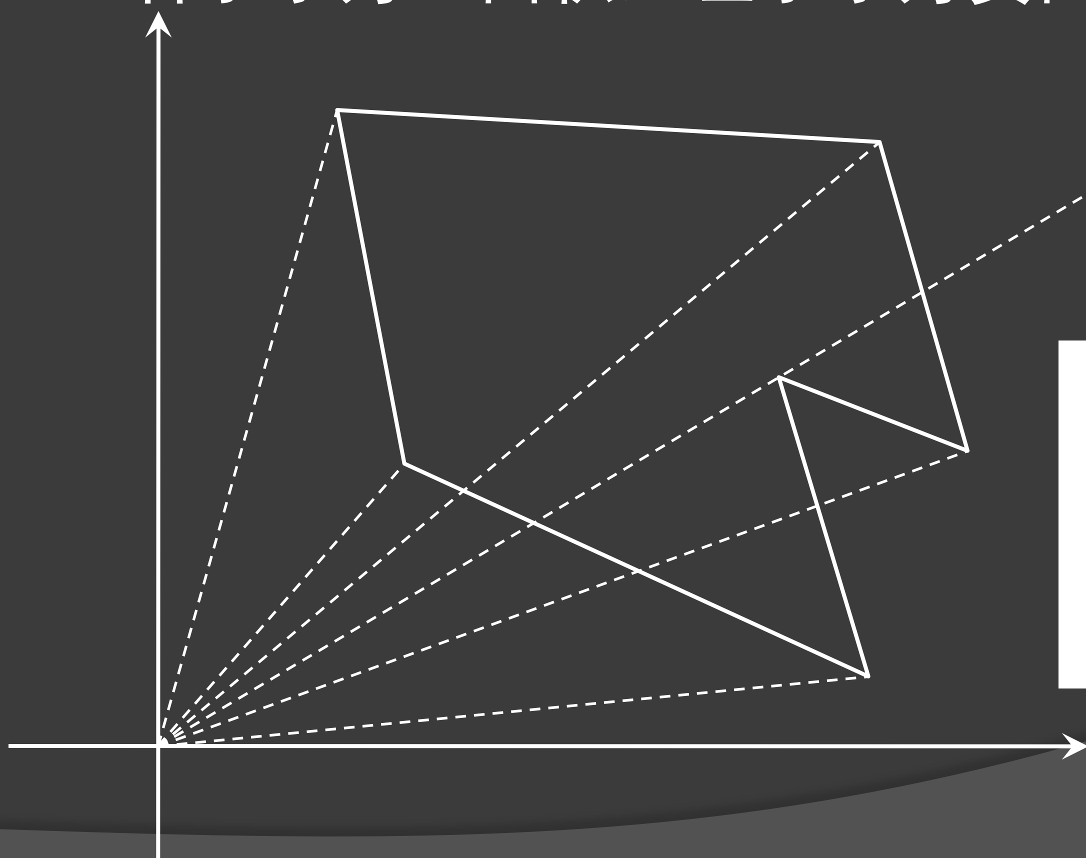


求凸多边形面积



求一般多边形面积

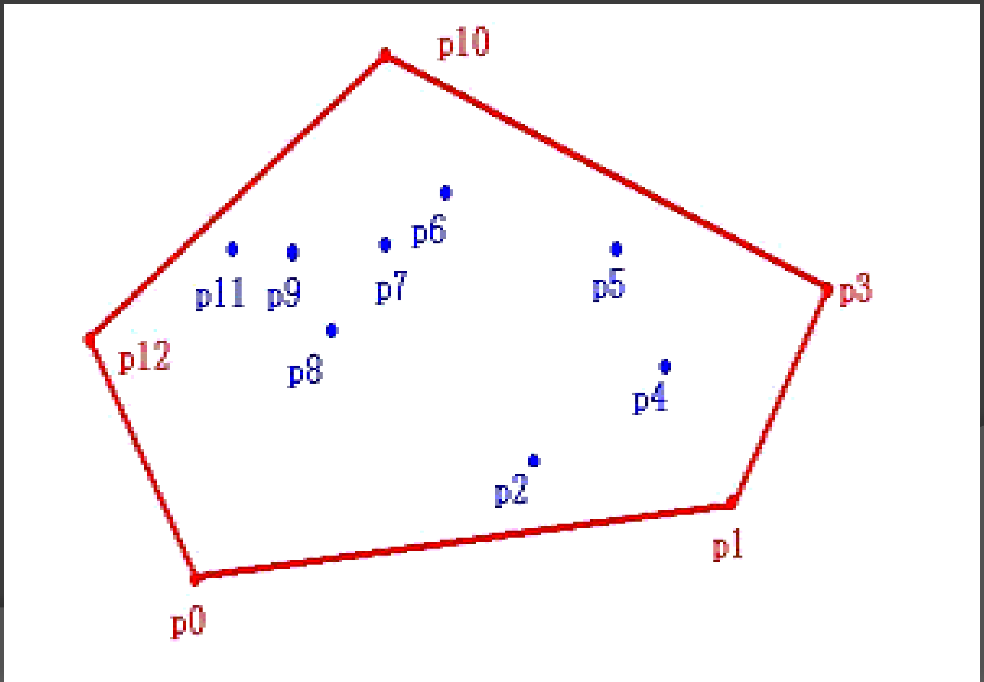
- 有向面积：叉乘结果保留符号
  - 右手系为正面积，左手系为负面积



$$\sum_{i=0}^{n-1} \begin{vmatrix} x_i & x_{i+1} \\ y_i & y_{i+1} \end{vmatrix} / 2$$

最常用几何模型——凸包

- 点集Q的凸包(convex hull)是指一个最小凸多边形，满足Q中的点或者在多边形边上或者在其内。下图中由红色线段表示的多边形就是点集Q={p0,p1,...p12}的凸包。



求凸包的Graham扫描法

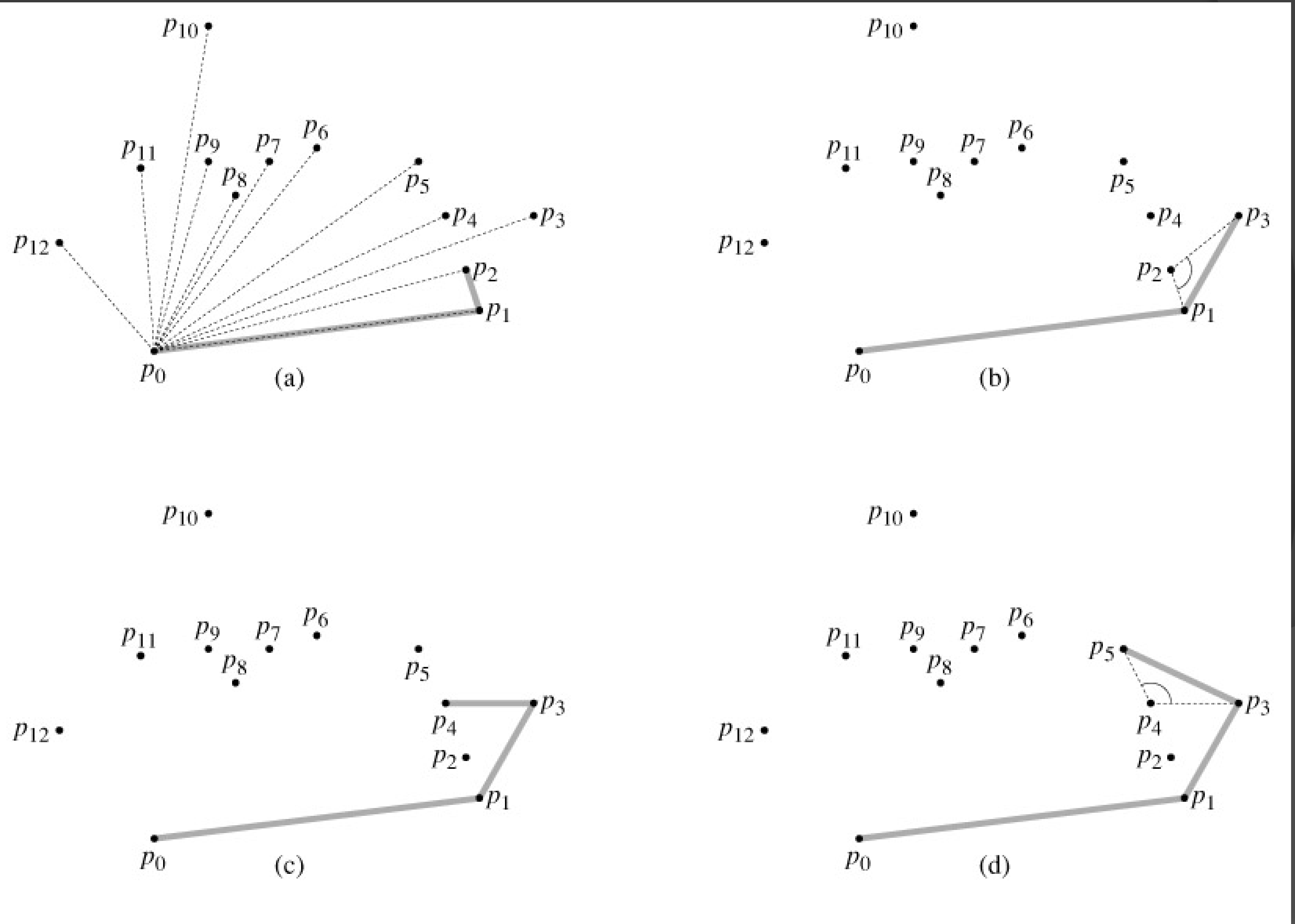
- 对于一个有三个或以上点的点集Q
- 令p0为Q中Y-X坐标排序下最小的点
  设<p1,p2,...pm>为对其余点按以p0为中心的极角逆时针排序所得的点集（如果有多个点有相同的极角，除了距p0最远的点外全部移除）
  压p0进栈S，压p1进栈S，压p2进栈S
  for(i=3;i<=m;++i){
      while(由S的栈顶元素的下一个元素、S的栈顶元素以及pi构成的折线段不拐向左侧){
          对S弹栈
          压pi进栈S
      }
  }
  return S

Graham Scan

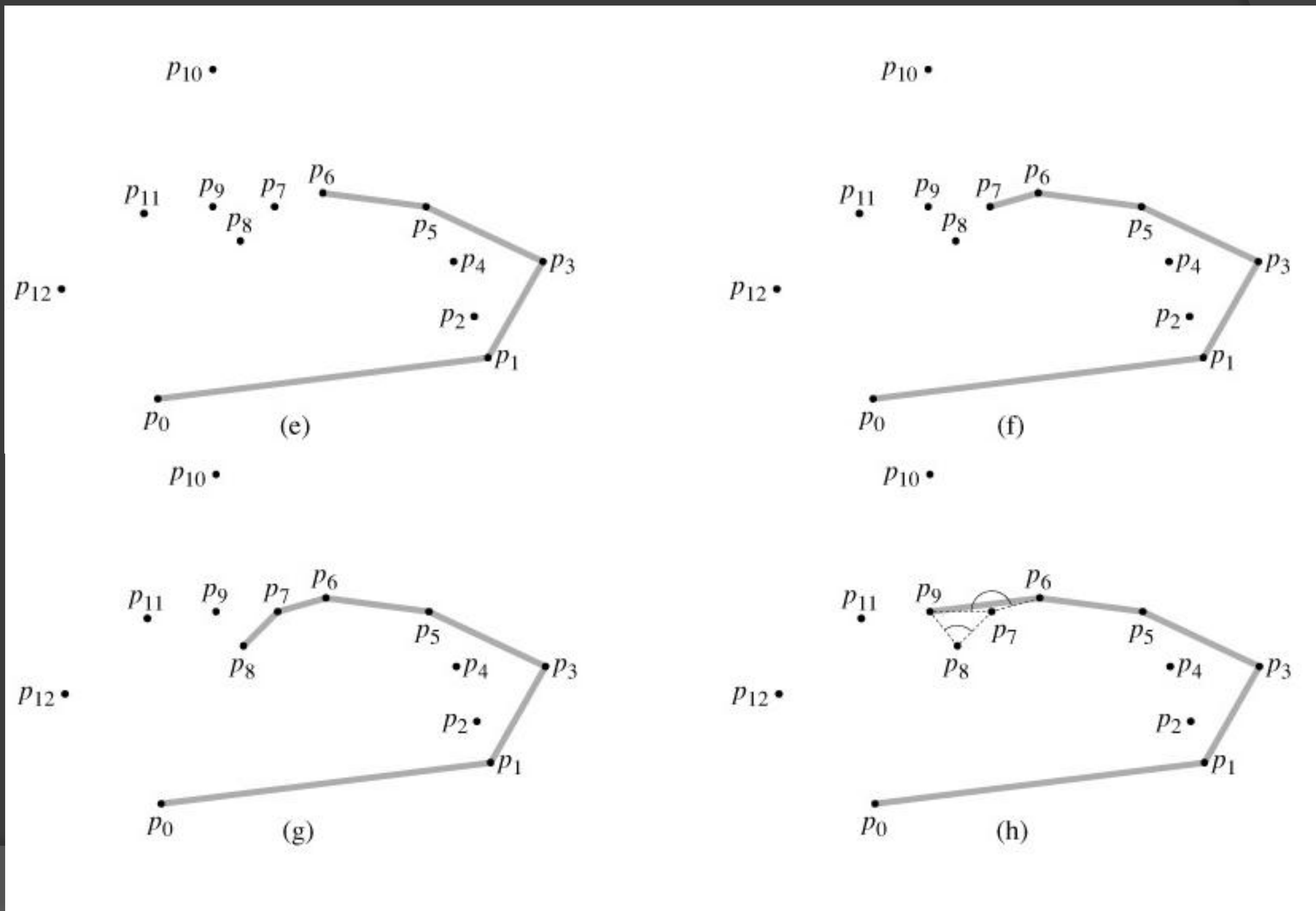
- 以叉积为依据排序，避免除法运算
- 极角==旋转方向

```
int cmp(Point a, Point b) {
  return sig(a.x*b.y-a.y*b.x) > 0;
}
```

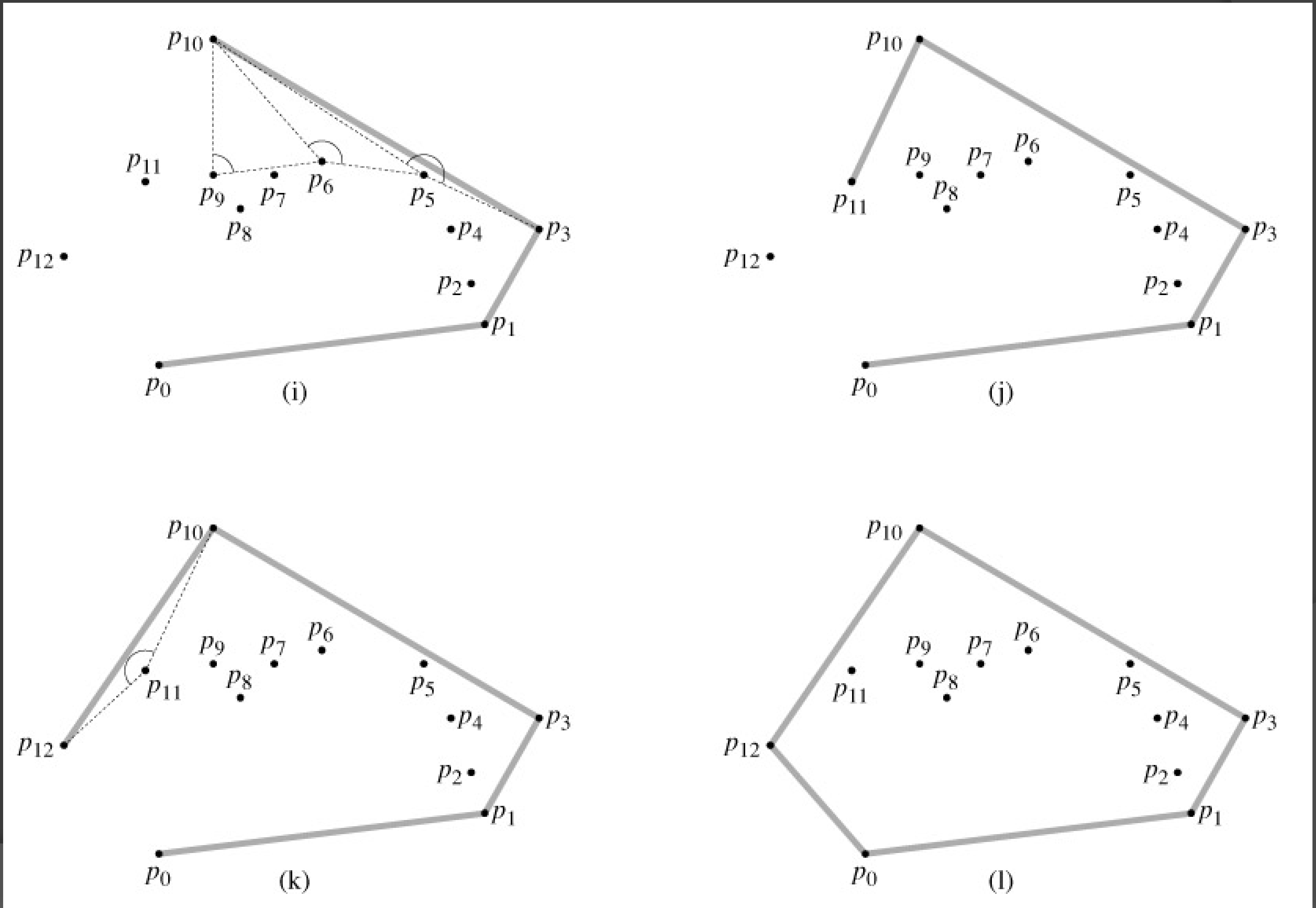
Graham Scan



Graham Scan



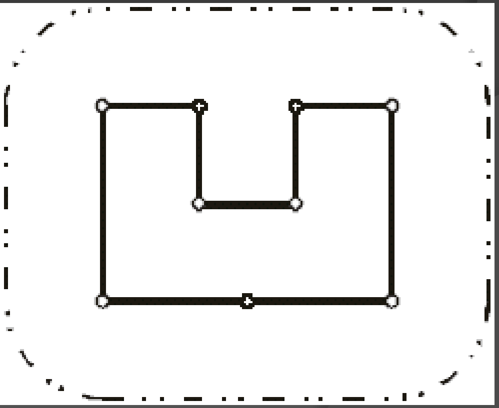
Graham Scan



凸包练习题

- http://acm.hdu.edu.cn/showproblem.php?pid=1392
  一堆树，用绳围起来，求最短绳长

- http://acm.tju.edu.cn/toj/showp.php?pid=2317
  多边形城堡修围墙，要求离城堡不能小于一定距离。求最节省材料的方案并求出围墙长。



TOJ1174 Triangle

- 求点集中面积最大的三角形
  - $1 \le n \le 50000$ and $-10^4 \le x_i, y_i \le 10^4$
- 如果枚举3个点再算面积，必然超时。
- 很显然的最大面积的三角形的三个顶点必然是这个点集的凸包上的点，因此先求出凸包。
- 如果求出凸包仍然枚举，一样会超时。

TOJ1174 Triangle

- 确定三个顶点不能使用$O(n^3)$的算法，会TLE。注意到面积具有二分的性质。在凸包上三角形的一个顶点顺时针移动的时候，凸包面积先增后减。
- 更快的算法：类似于rotating calipers…

GAME OVER!