GIS专业主干课 ：21905001

计算机图形学

Computer Graphics

林伟华

中国地质大学（武汉）信息工程学院

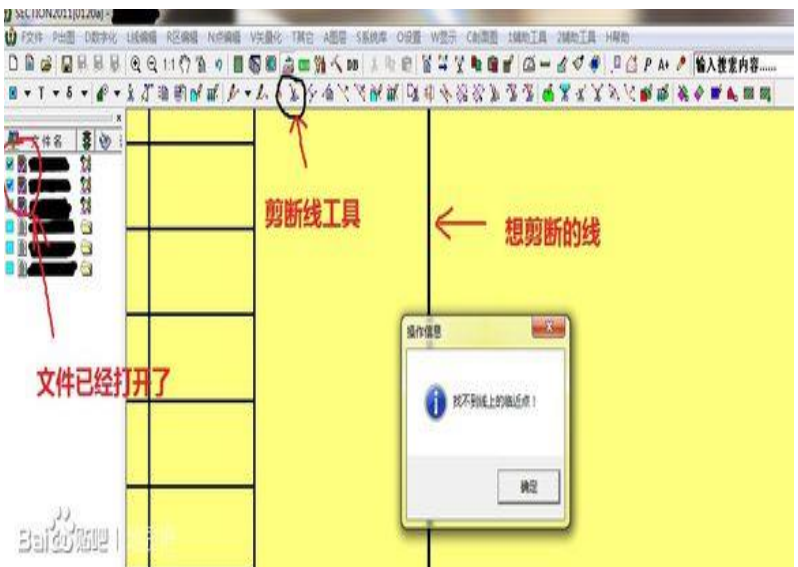
lwhcug@163.com

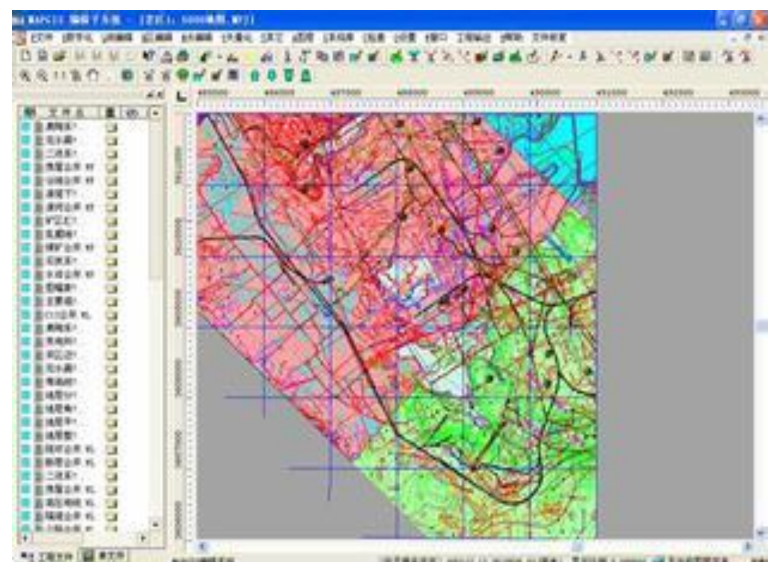
目录

- 矢量运算
- 点与线位置关系
- 多边形与多边形位置关系
- 交点计算
- 图形数据结构

GIS常见图形关系

- MapGIS开发
  - long _NearPnt(short ai, D_DOT *pxy);
  - long _NearLin(short ai, D_DOT *pxy); ……
- MapGIS应用
  - 剪断线
  - 地图裁剪……





九交集模型

Max J. Egenhofer

1865 THE UNIVERSITY OF
MAINE

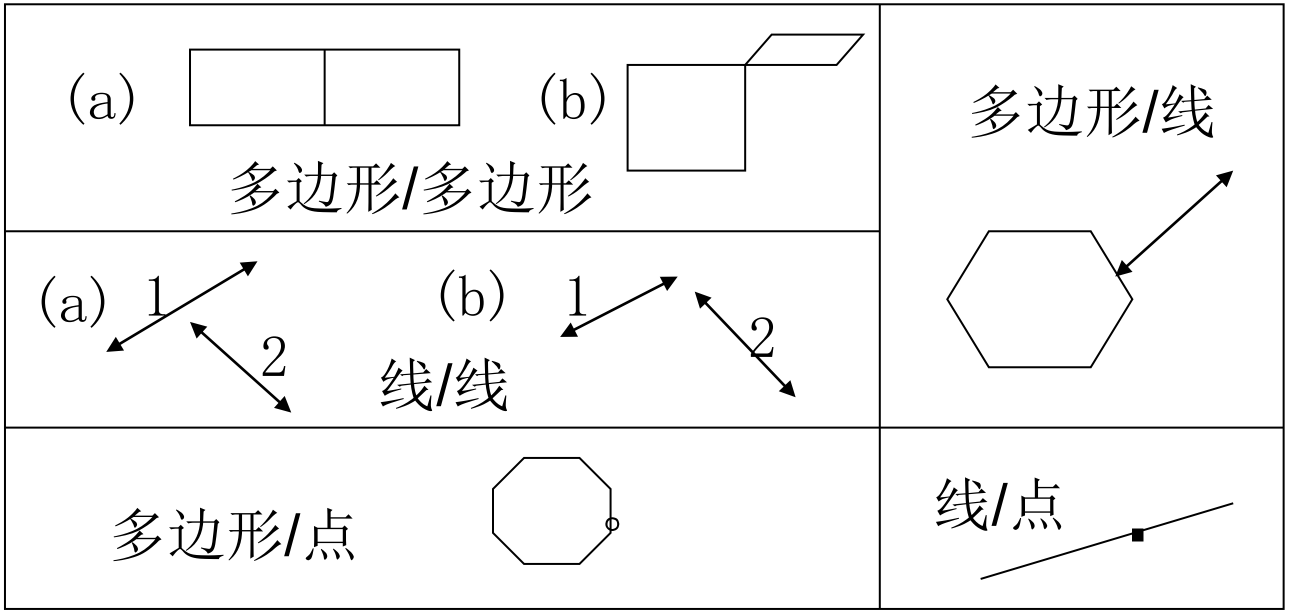
内涵：

设有空间对象A、B，$B(A)$、$B(B)$表示A、B的边界，$I(A)$、$I(B)$表示A、B的内部，$E(A)$、$E(B)$表示A、B的外部或余，二者之间的拓扑关系可用表示：

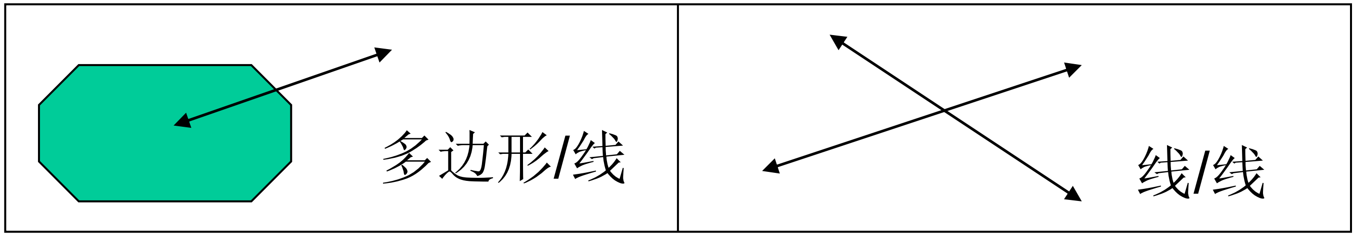
$$\begin{bmatrix} I(A)\cap I(B) & I(A)\cap B(B) & I(A)\cap E(B) \\ B(A)\cap I(B) & B(A)\cap B(B) & B(A)\cap E(B) \\ E(A)\cap I(B) & E(A)\cap B(B) & E(A)\cap E(B) \end{bmatrix}$$

Egenhofer M.. A Model for Detailed Binary Topological Relationships[J]. Geomatic, 1993, 47(3): 261-273.

九交集模型

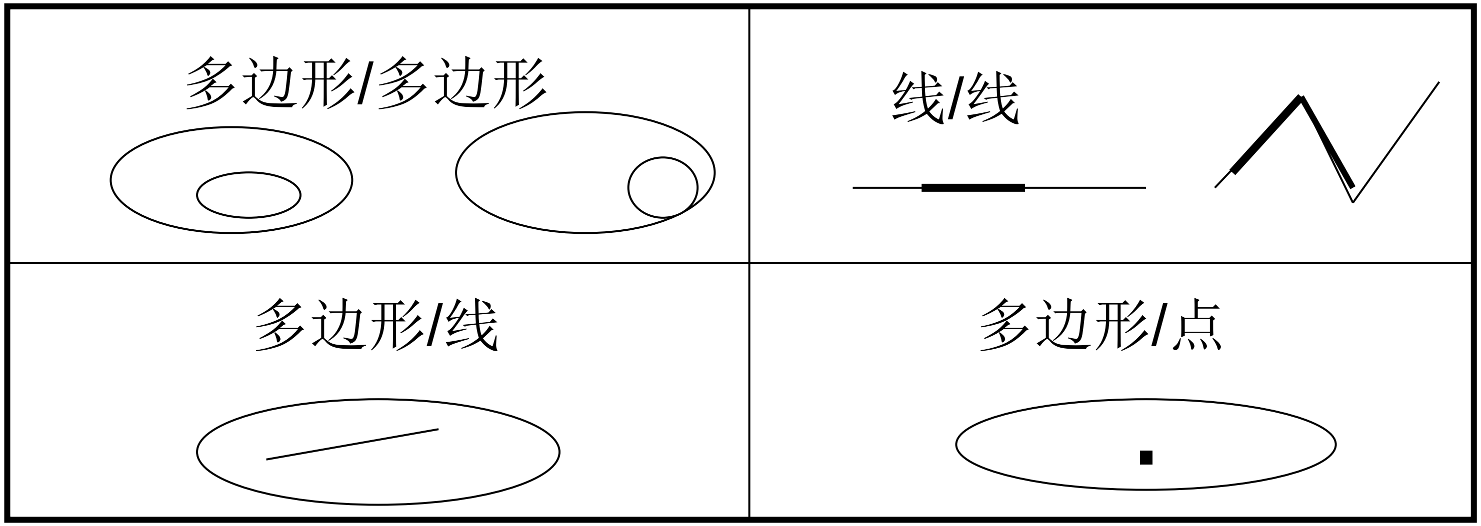


相接关系示例

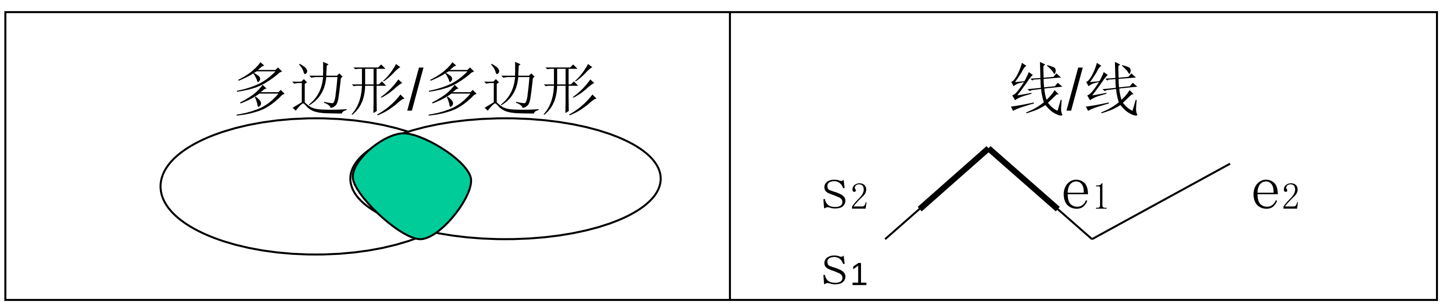


相交关系示例

九交集模型



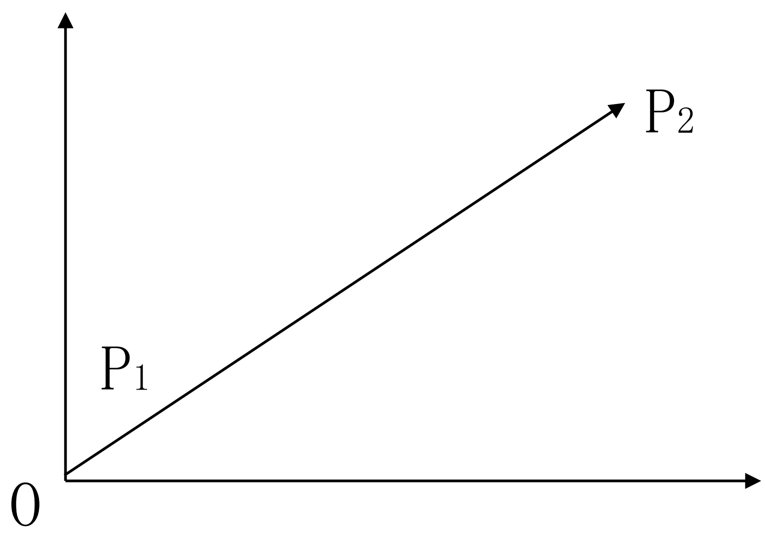
真包含关系示意图



叠置关系示例

矢量的概念

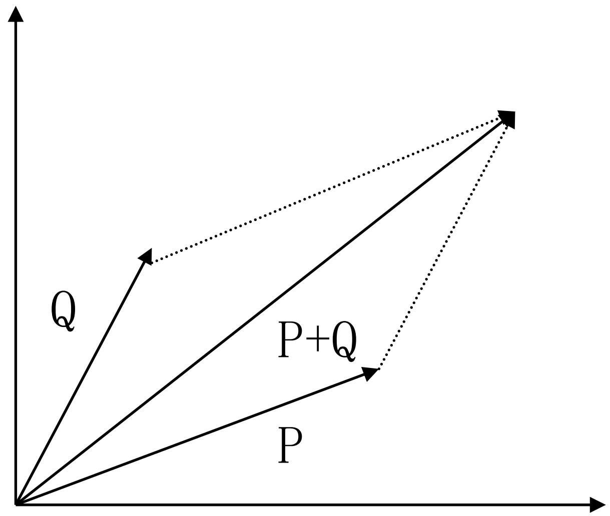
- 一条线段的端点是有次序之分的，这种线段称为有向线段。
- 如果有向线段P1P2的起点P1在坐标原点，我们可以把它成为矢量P2。



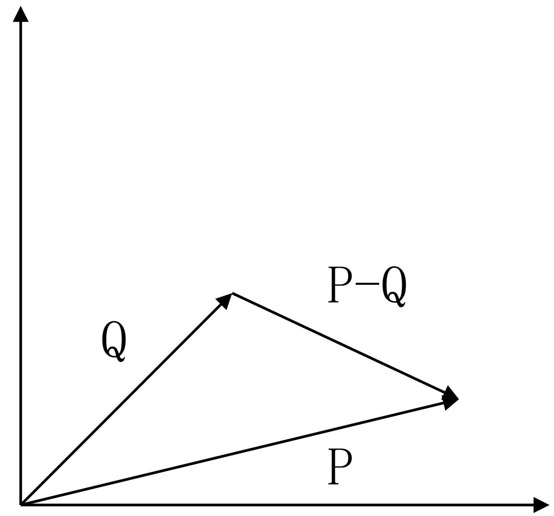
矢量的概念

矢量加减法

◆ 设二维矢量P=（x1,y1），Q=(x2,y2),则：



矢量加法



矢量减法

向量a和向量b，向量的点积为：a•b=｜a｜｜b｜cosθ
向量a和向量b，向量的叉积为：｜a×b｜=｜a｜｜b｜sinθ

矢量乘法

- **向量a和向量b，向量的点积为：**

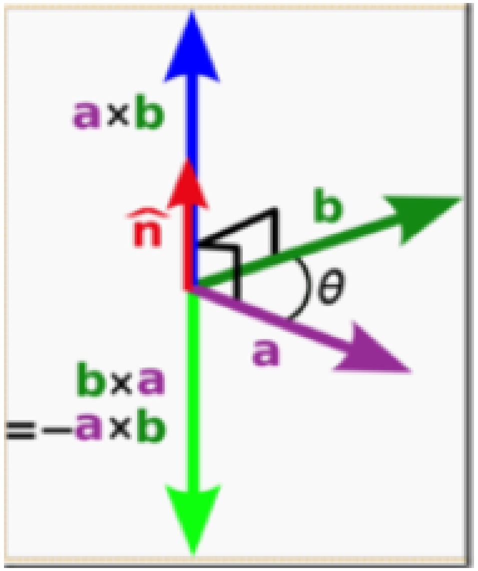
  $a \cdot b = |a|\ |b| \cos\theta$

  点积为一标量，物理含义为力在位移上所做的功。

- **向量a和向量b，向量的叉积为：**

  $a \times b = |a|\ |b| \sin\theta$

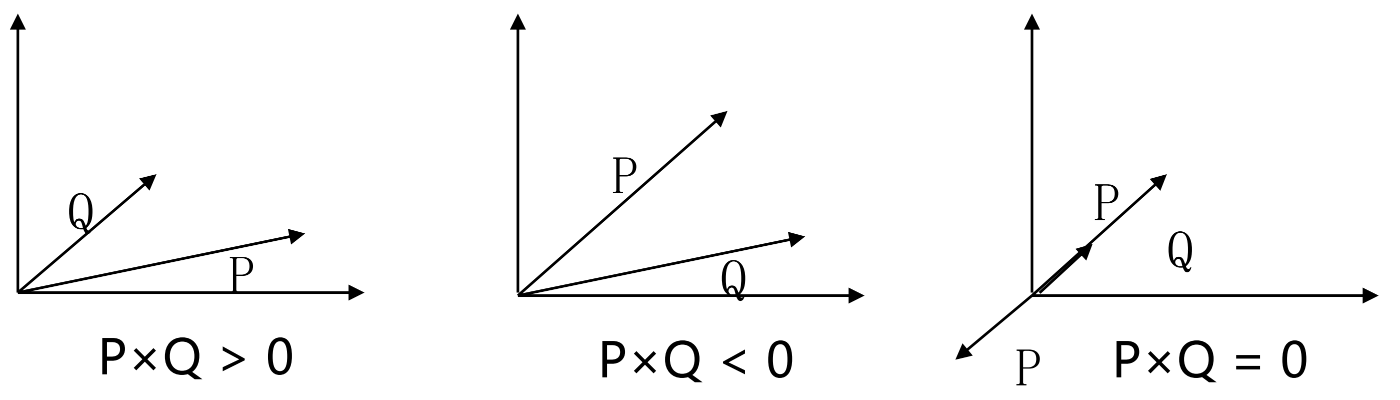
  叉积是一个向量，并且方向与这两个向量的和垂直。



矢量叉积

设二维矢量P=（x1,y1），Q=(x2,y2),其矢量叉积有：

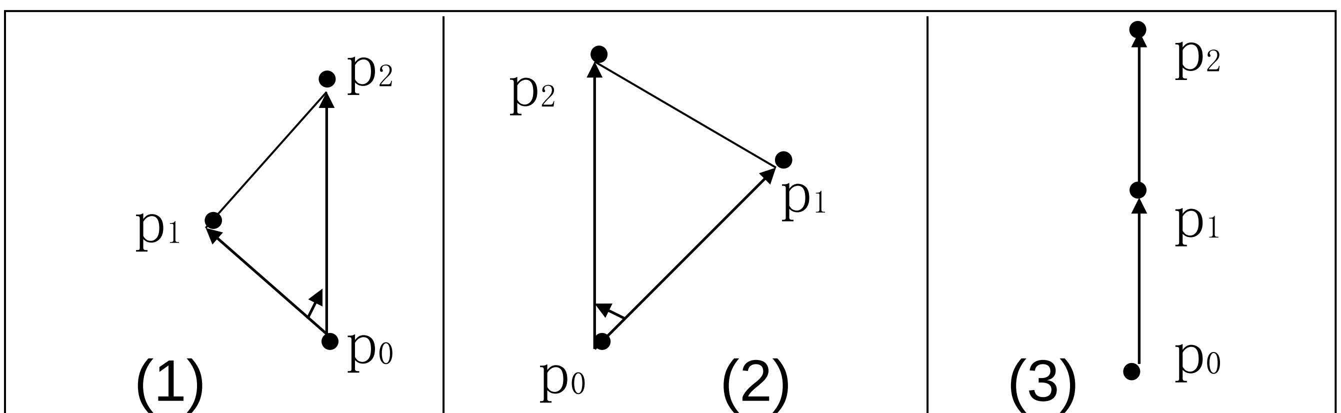
- **取模**是由(0,0)、P、Q和PQ所组成的平行四边形的带符号的面积。
- P×Q=x1·y2-x2·y1 是一个标量，并且
  P×Q=-(Q×P) 和 P×（-Q）=-(P×Q)
- P×Q的符号判断两矢量相互之间的顺逆时针关系：
  (1)若P×Q > 0，则P在Q的顺时针方向；
  (2)若P×Q < 0，则P在Q的逆时针方向；
  (3)若P×Q = 0，则P与Q共线，但可能同向也可能反向。



折线段拐向

线段p0p1和p1p2，通过计算(p2 - p0)×(p1 - p0)的符号确定折线段的拐向：

- （1）若(p2 - p0) × (p1 - p0) > 0,则p0p1在p1点拐向右侧后得到p1p2。
- （2）若(p2 - p0) × (p1 - p0) < 0,则p0p1在p1点拐向左侧后得到p1p2。
- （3）若(p2 - p0) × (p1 - p0) = 0,则p0、p1、p2三点共线。



点在线上

设点为Q，线段为P1P2 ，判断点Q在该线段上的依据：
(Q - P1) × (P2 - P1) = 0 ，且 Q 在以 P1，P2为对角顶点的矩形内。

- 前者保证Q点在直线P1P2上。
- 后者是保证Q点不在线段P1P2的延长线或反向延长线上。

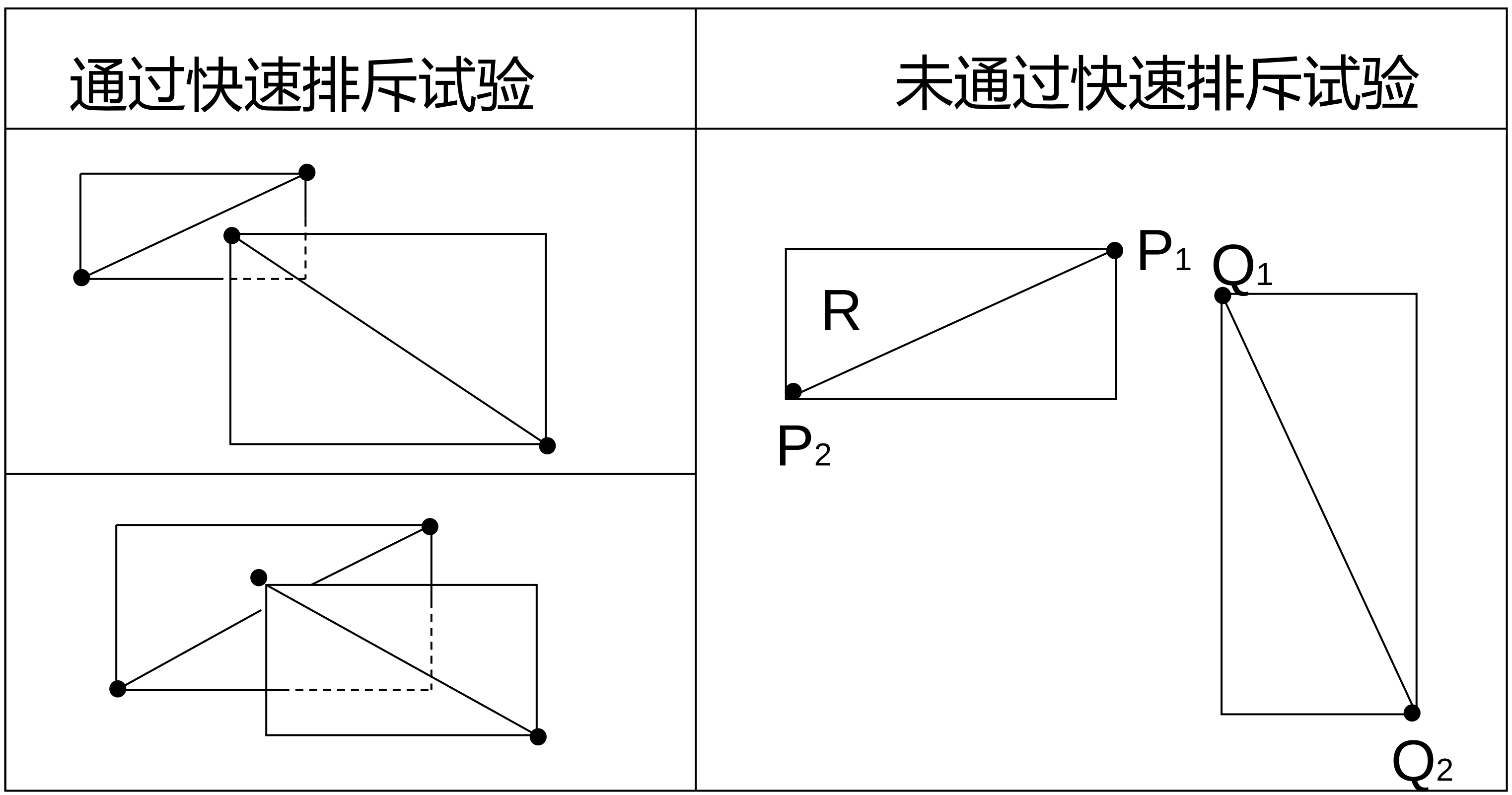
```
bool  ON-SEGMENT(pi,pj,pk)
{  if min(xi,xj)<=xk<=max(xi,xj)  and
        min(yi,yj)<=yk<=max(yi,yj)
        return true;
    else
        return false;
}
```

- 另外，水平和垂直线段，min(xi,xj)<=xk<=max(xi,xj)和min(yi,yj)<=yk<=max(yi,yj)两个条件须同时满足才返真。

两线段相交

判断方法：

- (1)快速排斥试验
- (2)跨立试验



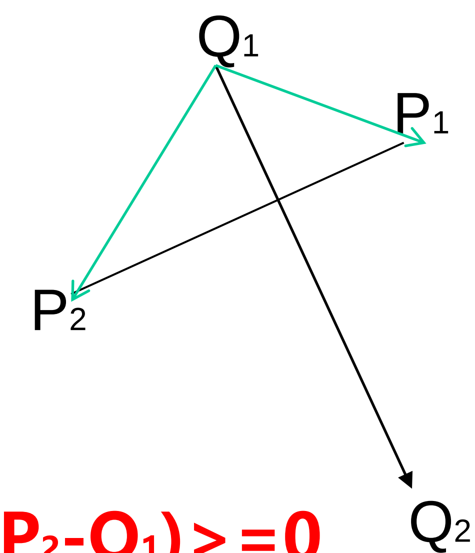
两线段相交

跨立试验判断：两线段相交，则两线段必然相互跨立对方

- 若P1P2跨立Q1Q2 ，即矢量(P1 - Q1)和(P2 - Q1)位于矢量(Q2 - Q1)的两侧。
- $(P1-Q1)\times(Q2-Q1)*(P2-Q1)\times(Q2-Q1)<0$。可写成：$(P1-Q1)\times(Q2-Q1)*(Q2-Q1)\times(P2-Q1)>0$。

(1) 当$(P1-Q1)\times(Q2-Q1)=0$时，说明(P1-Q1)和(Q2-Q1)共线，但是因为已经通过快速排斥试验，所以P1一定在线段Q1Q2上；

(2) 当$(Q2-Q1)\times(P2-Q1)=0$ 时，说明P2一定在线段 Q1Q2上。



跨立判断：$(P_1-Q_1)\times(Q_2-Q_1)*(Q_2-Q_1)\times(P_2-Q_1)>=0$

线段与直线相交

线段P1P2和直线Q1Q2相交，则P1P2跨立Q1Q2：

- (P1-Q1)×(Q2-Q1)*(Q2-Q1)×(P2-Q1)>=0

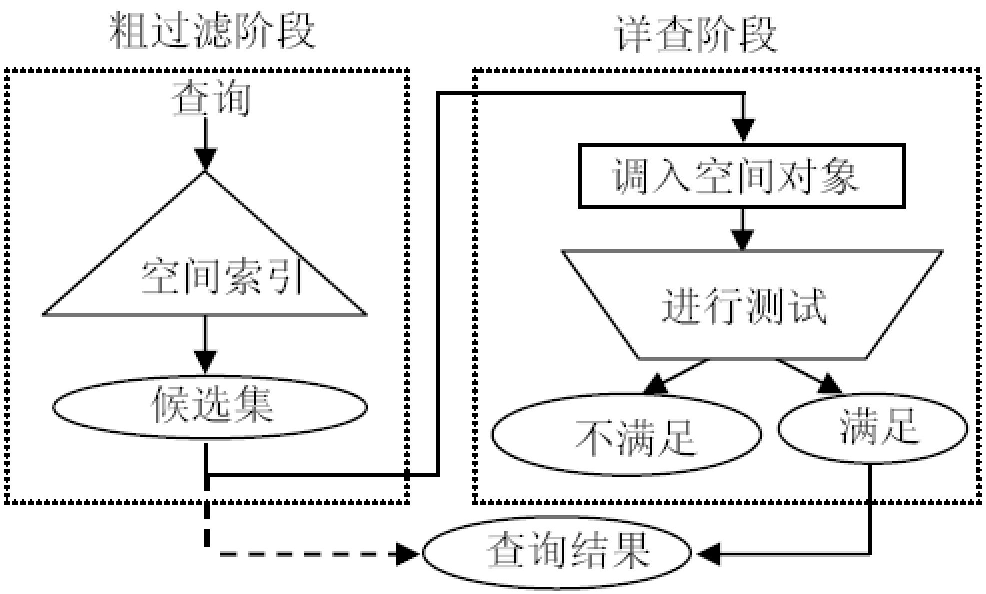
- 注意：不需进行快速排斥实验

思考：

1、点在矩形内的判断？

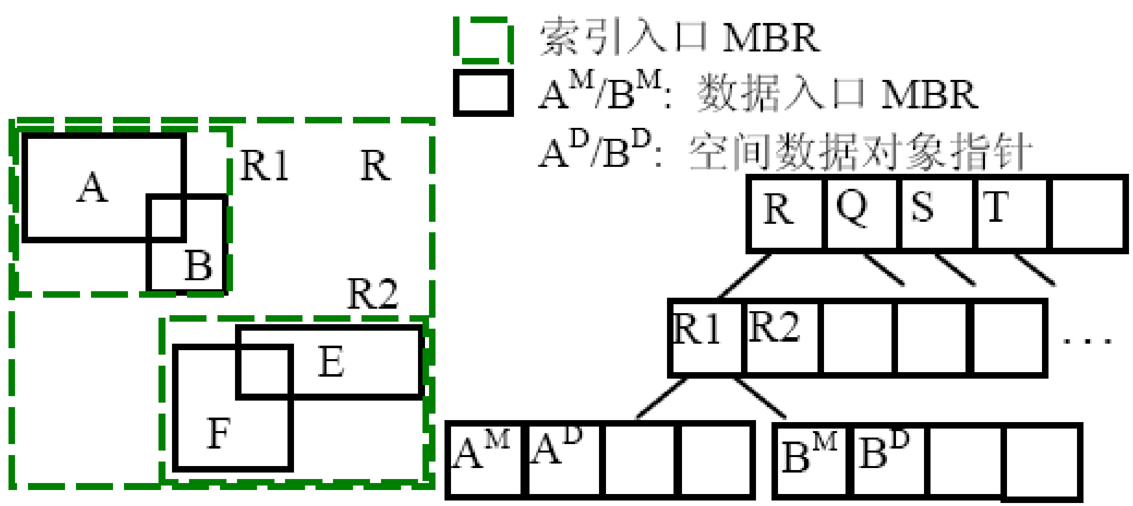
2、线段、折线、矩形在矩形内的判断？

快速排斥实验应用



空间数据检索机制

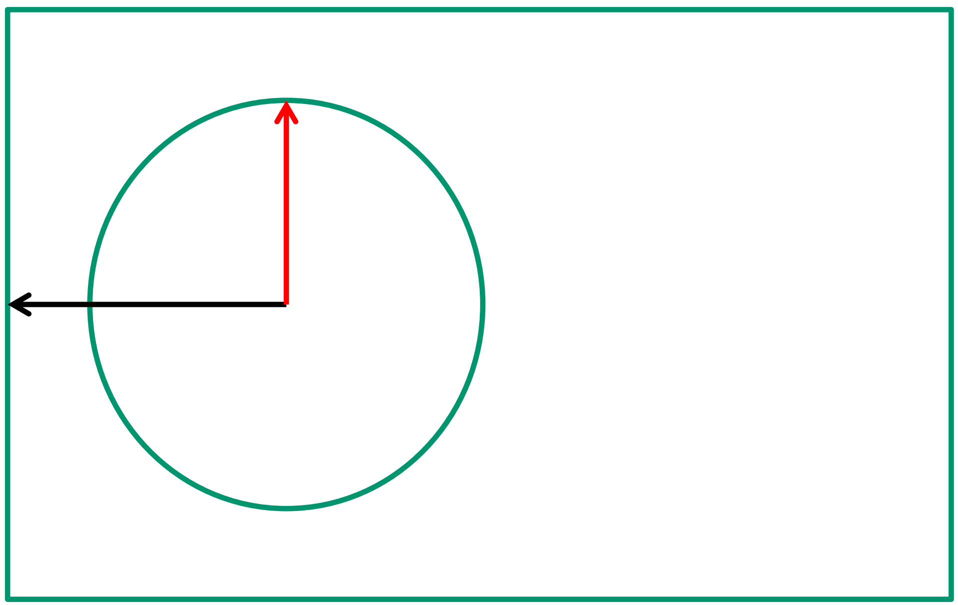
R-tree索引结构



圆在矩形中

判断条件：

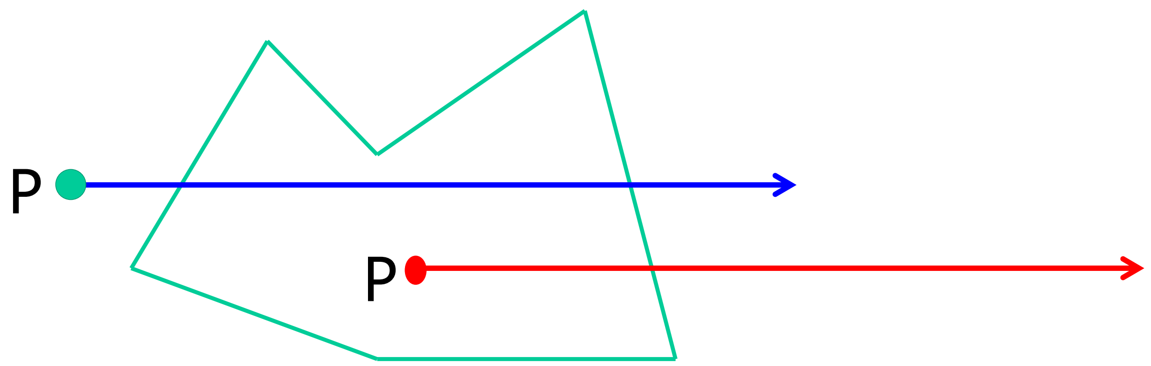
- 圆心在矩形中
- 圆的半径小于等于圆心到矩形四边的距离的最小值



点在多边形内

判断条件：

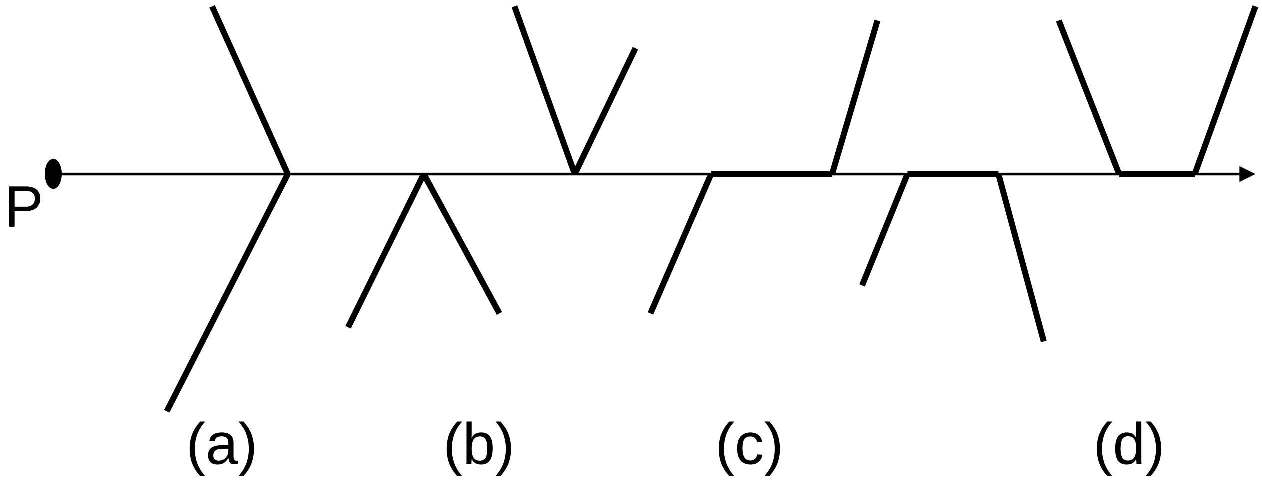
- 以点P为端点，向右方作射线L；
- 考虑沿着L从无穷远处开始自右向左移动，遇到和多边形的第一个交点的时候，进入到了多边形的内部，遇到第二个交点的时候，离开了多边形，……
- 当L和多边形的交点数目是奇数时，P在多边形内，是偶数时，P在多边形外。



点在多边形内

考虑特殊情况：

- 图(a)中，L和多边形的顶点相交，这时候交点只能计算一个；
- 图(b)中， L和多边形顶点的交点不应被计算；
- 图(c)和(d) 中，L和多边形的一条边重合，这条边应该被忽略不计。



思考：图(a)与图(b)如何区分？

点在多边形内

算法思路：

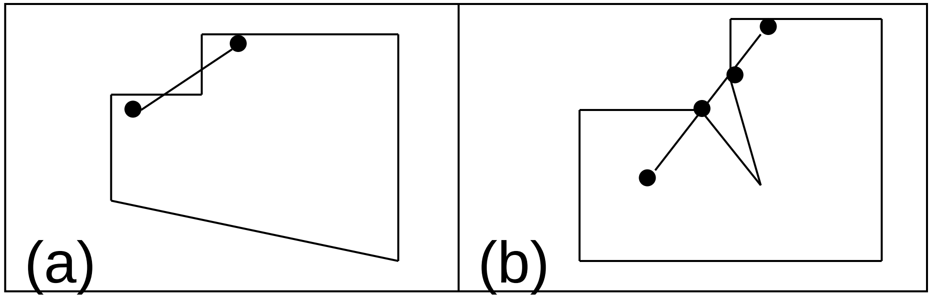
```
count ← 0;
        以P为端点，作从右向左的射线L;
        for 多边形的每条边s
         do if P在边s上
              then return true;
            if s不是水平的
              then if s的一个端点在L上
                     if 该端点所属的两边在射线L异侧
                       then count ← count+1
                   else if s和L相交
                     then count ← count+1;
        if count mod 2 = 1
          then return true;
        else return false;
```

- **作射线L的方法是：设P‘的纵坐标和P相同，横坐标为正无穷大，则P和P’就确定了射线L；**
- **算法的时间复杂度为O(n)。**

线段在多边形内

判断条件：

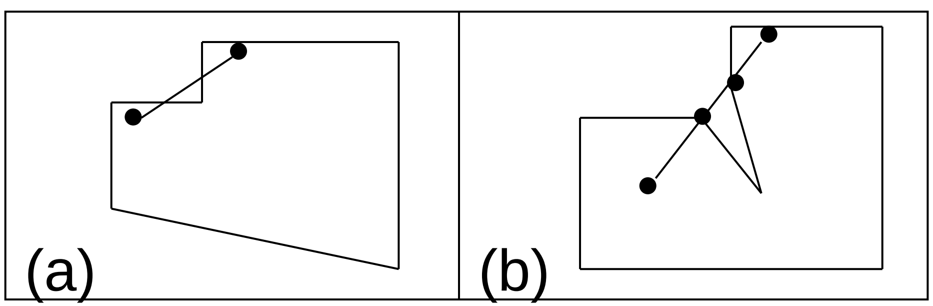
- **凸多边形：线段的两个端点都在多边形内或在多边形边上。**
- **凹多边形：**
  - ◆线段和多边形的所有边都不内交（两线段内交是指两线段相交且交点不在两线段的端点）。
  - ◆若线段和多边形交于线段的两端点，可判断线段是否在多边形内；
  - ◆若多边形的某个顶点和线段相交，还必须判断两相邻交点之间的线段是否在多边形内部（反例见图b)。



线段在多边形内

两相邻交点之间的线段是否在多边形内部：

- 求出所有和线段相交的多边形的顶点，然后按照X-Y坐标排序(X坐标小的排在前面，对于X坐标相同的点，Y坐标小的排在前面，这样相邻的两个点就是在线段上相邻的两交点。
- 如果任意相邻两点的中点也在多边形内，则该线段一定在多边形内。



线段在多边形内

任意相邻两点的中点也在多边形内，则该线段一定在多边形内：

命题1:如果线段和多边形的两相邻交点P1 ，P2的中点P’也在多边形内，则P1, P2之间的所有点都在多边形内。

证明：

假设P1,P2之间含有不在多边形内的点，不妨设该点为Q，在P1, P’ 之间，因为多边形是闭合曲线，所以其内外部之间有界，而P1属于多边行内部，Q属于多边性外部，P’ 属于多边性内部，P1-Q-P’ 完全连续，所以P1Q和QP’ 一定跨越多边形的边界，因此在P1,P'之间至少还有两个该线段和多边形的交点，这和P1P2是相邻两交点矛盾，故命题成立。证毕。

线段在多边形内

推论：设多边形和线段PQ的交点依次为P1,P2,……Pn，其中Pi和Pi+1是相邻两交点，线段PQ在多边形内的充要条件是：P，Q在多边形内且对于i =1, 2,……, n-1，Pi ,Pi+1的中点也在多边形内。

算法思路1：线段端点在多边形内，计算所有的交点，判断中点是否在多边形内。

算法思路2：

- ◆线段端点在多边形内，判断线段和多边形的边是否内交。
- ◆若线段和多边形的某条边内交则线段一定在多边形外，则FALSE。
- ◆若线段和多边形的每一条边都不内交，如果线段和多边形无交点，则TRUE。
- ◆若线段和多边形的每一条边都不内交，则线段和多边形的交点一定是线段的端点或者多边形的顶点，判断中点是否在多边形内。

线段在多边形内

算法思路2：

```
if 线段PQ的端点不都在多边形内
then return false;
else
  点集pointSet初始化为空;
for 多边形的每条边s
do if 线段的某个端点在s上
   then 将该端点加入pointSet;
   else if s的某个端点在线段PQ上
        then 将该端点加入pointSet;
        else if s和线段PQ相交 /*这时候已经可以肯定是内交了*/
             then return false;
                else
                将pointSet中的点按照X-Y坐标排序;
                for pointSet中每两个相邻点pointSet[i],
                                        pointSet[i+1]
                  do if pointSet[i] , pointSet[ i+1] 的
                        中点不在多边形中
                     then return false;
                     else return true
```

因交点数目肯定远小于多边形的顶点数目n，可忽略不计。该算法的时间复杂度也是O(n)。

线段在多边形内

思考：

- **判断折线是否在多边形内？**

  时间复杂度是$O(m \times n)$

- **判断多边形是否在多边形内？**

  时间复杂度是$O(m \times n)$

- **判断圆是否在多边形内？**

  计算圆心到每条边最小距离

凹、凸多边形判断

判断方法1：

依次判断多边形顶点顺时针方向（或逆时针方向）两边拐向是否一致。

判断方法2：

- ◆依次平移多边形每个顶点到原点
- ◆顺时针方向旋转该顶点关联的一边（每次选择的边方向相同）
- ◆如所有下一顶点在X轴上方，则为凸多边形，否则为凹多边形。

点、线段在圆内

判断点在圆内：

计算圆心到该点的距离，如果小于等于半径则该点在圆内。

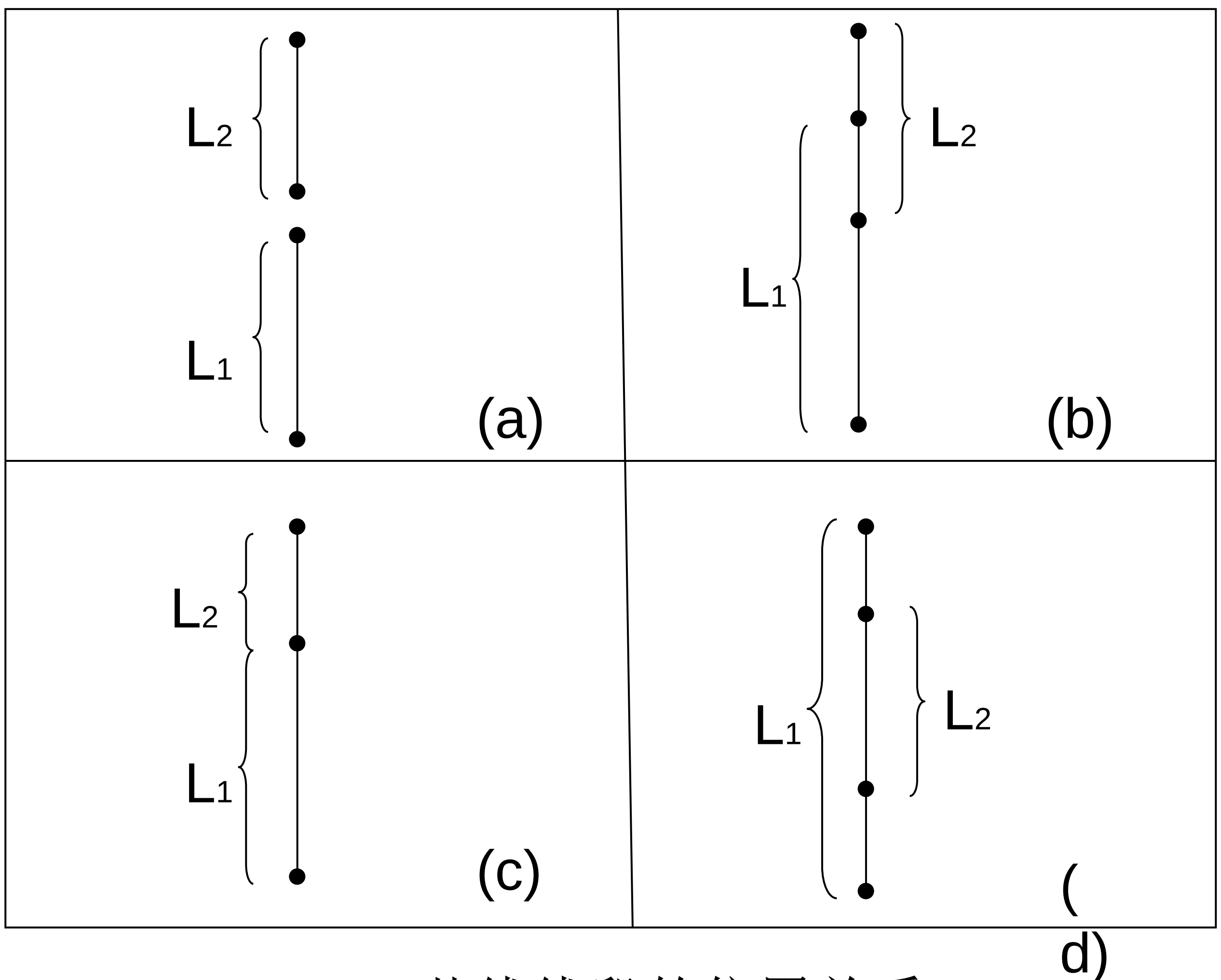
判断线段、折线、矩形、多边形在圆内：

因**圆是凸集**，所以只要判断是否每个顶点都在圆内即可。

判断圆在圆内：

设两圆为O1,O2，半径分别为r1, r2，要判断O2是否在O1内。先比较r1，r2的大小，如果r1<r2则O2不可能在O1内；否则如果两圆心的距离大于r1 - r2 ，则O2不在O1内；否则O2在O1内。

共线线段的交点



共线线段的位置关系

线段与线段的交点

算法思路：

设一条线段为L0=P1P2，另一条线段或直线段L1=Q1Q2，计算L0和L1的交点：

- 1.首先判断L0和L1是否相交，如果不相交则没有交点，否则说明L0和L1一定有交点。
- 2.如果P1和P2横坐标相同，即L0平行于Y轴
  a)若L1也平行于Y轴
  b) 若L1不平行于Y轴，
- 3.如果P1和P2横坐标不同，但是Q1和Q2横坐标相同
- 4.如果P1和P2纵坐标相同，即L0平行于X轴
  a) 若L1也平行于X轴
  b) 若L1不平行于X轴
- 5.如果P1和P2纵坐标不同，但是Q1和Q2纵坐标相同
- 6.剩下的情况就是L1和L0的斜率均存在且不为0的情况
  a) 计算出L0的斜率K0，L1的斜率K1；
  b) 如果K1=K2
  c) 联立两直线的方程组可以解出交点

线段与圆的交点

算法思路：

设圆心为O，圆半径为r，直线(或线段）L上的两点为P1, P2。

- 1.如果L是线段且P1，P2都包含在圆O内，则没有交点；否则进行下一步。
- 2.如果L平行于Y轴，

  a) 计算圆心到L的距离dis；

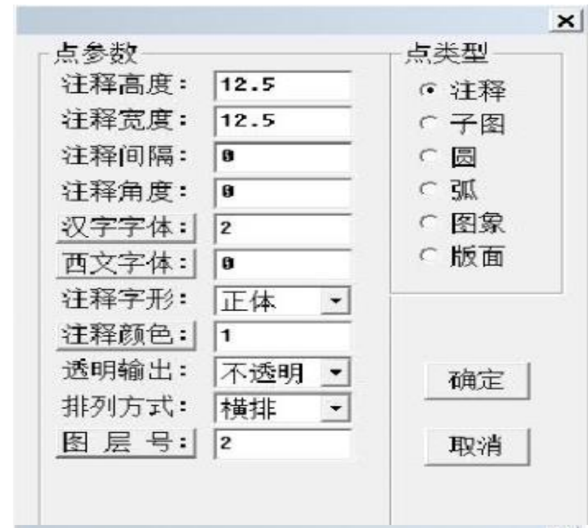
  b) 如果dis>r则L和圆没有交点；

  c) 利用勾股定理，可以求出两交点坐标，但要注意考虑L和圆的相切情况。
- 3.如果L平行于X轴，做法与L平行于Y轴的情况类似；
- 4.如果L既不平行X轴也不平行Y轴，可以求出L的斜率K，然后列出L的点斜式方程，和圆方程联立即可求解出L和圆的两个交点；
- 5.如果L是线段，对于2，3，4中求出的交点还要分别判断是否属于该线段的范围内。

点参数

PNT_INFO   inf;

- switch(inf.type)
- {
-     case PNT_NOTE: …   break;   //处理字符串
-     case PNT_SUB: …   break;   //处理子图
-     case PNT_CIR: …   break;   //处理圆
-     case PNT_ARC: …   break;   //处理弧
-     case PNT_IMAGE: … break;   //处理图象
-     case PNT_TEXT: …   break;   //处理文本
-         default: break;
- }
- …

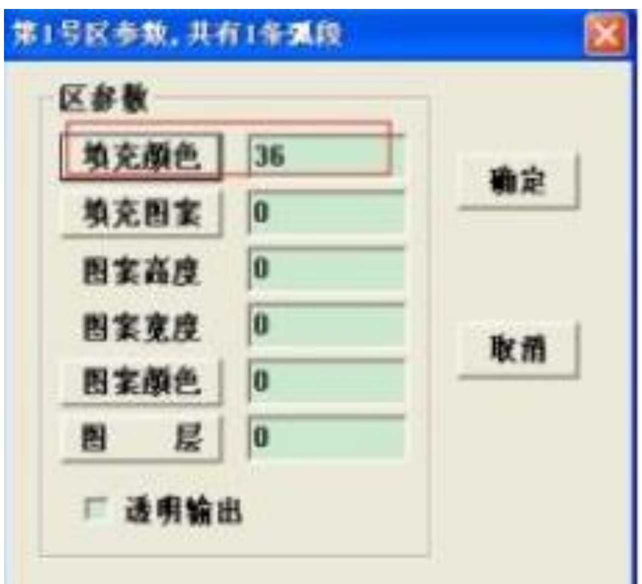


线-区参数

LIN_INFO inf;

- memset(&inf,0,sizeof(LIN_INFO));
- inf.ltp=1;   //线型号
- inf.lclr=7;   //线颜色
- inf.lw=1;   //线宽

REG_INFO inf;

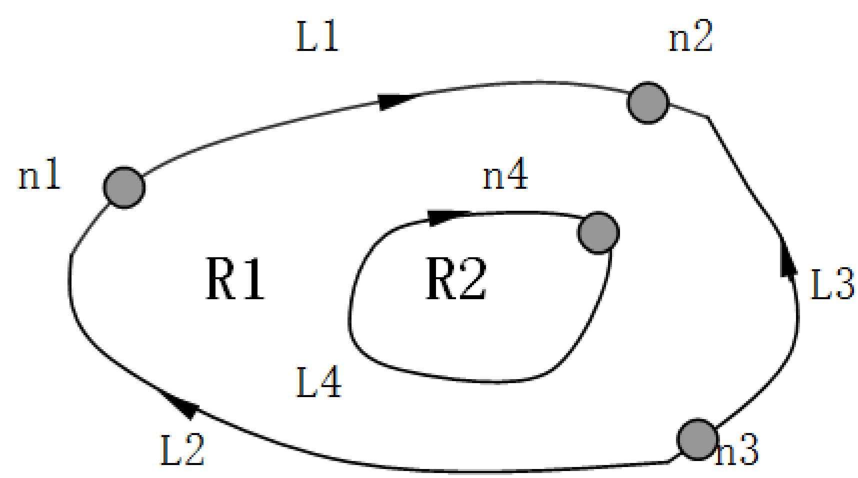


点-线-区拓扑结构

LIN_TOP：L2的拓扑结构

NOD_TOP： n1的拓扑结构

REG_TOP：R1的拓扑结构



n3	n1	0	R1	0	0

起始结点 终止结点 左区 右区

L1	L2

d1+d2+d3+d4+2	L1	-L3	L2	0	L4

思考

1、写出点到线段的最小距离算法。

谢　谢！