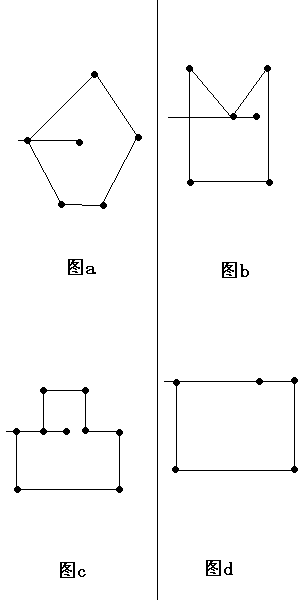
**判断点是否在多边形中：**

　　判断点P是否在多边形中是计算几何中一个非常基本但是十分重要的算法。以点P为端点，向左方作射线L，由于多边形是有界的，所以射线L的左端一定在多边形外，考虑沿着L从无穷远处开始自左向右移动，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，……所以很容易看出当L和多边形的交点数目C是奇数的时候，P在多边形内，是偶数的话P在多边形外。

　　但是有些特殊情况要加以考虑。如图下图(a)(b)(c)(d)所示。在图(a)中，L和多边形的顶点相交，这时候交点只能计算一个；在图(b)中，L和多边形顶点的交点不应被计算；在图(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。如果L和多边形的一条边重合，这条边应该被忽略不计。



　　为了统一起见，我们在计算射线L和多边形的交点的时候，1。对于多边形的水平边不作考虑；2。对于多边形的顶点和L相交的情况，如果该顶点是其所属的边上纵坐标较大的顶点，则计数，否则忽略；3。对于P在多边形边上的情形，直接可判断P属于多边行。由此得出算法的伪代码如下：  
  
    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 该端点是s两端点中纵坐标较大的端点  
                   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;  
  
  
　　其中做射线L的方法是：设P'的纵坐标和P相同，横坐标为正无穷大（很大的一个正数），则P和P'就确定了射线L。

　　判断点是否在多边形中的这个算法的时间复杂度为O(n)。