将多边形顶点根据其与点q的连接线段的角度排序，依照此顺序扫描这些顶点。以q出发的任意一条射线开始，求出此射线与P的交点数量，然后旋转此射线，经过多边形顶点时，根据该点相关的两条边的位置确定交点数的变化情况。两条边 左侧 +2
右侧 −2
两侧不变

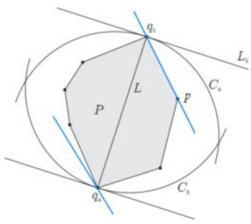
0420

1 设 P 是包围在给定矩形 R 中的一个简单多边形，q 为 R 中任意一点，设计高效算法寻找连接 q 和 R 外部一点的线段，使得该线段与 P 相交的边的数量最少

算法：
在 R 外找点两个不同的点 s1，s2 使得线段 qs1 和 qs2 的平行于多边形的一边（两条线段属于同一条直线但不重合）判断这两条线段和多边形每条边是否相交从而记录相交的边数。对多边形的每一条边重复这样的操作，最后选出相交边最少的线段。

2 给定平面上一组点，已知每个点的坐标，求最远点对之间的距离，即点集的直径。（不得穷举，文献查阅，然后用自己的语言进行算法思想的描述，包括时间复杂性分析）

算法一：穷举法，和求最近点对之间的距离相似，对每个点依次求它与其余$n-1$个点对的距离，最后选出距离最远的两个点。时间复杂度：$O(n^2)$
算法二：利用几何性质，先求出这组点的凸包，时间复杂度为$O(n\log n)$。之后再对这个凸包利用旋转卡壳法寻找这个凸包的直径。具体方法为枚举凸包上的所有边，对每一条边找出凸包上离该边最远的顶点，计算这个顶点到该边两个端点的距离，并记录最大的值。当逆时针枚举边的时候，最远点的变化也是逆时针的，这样就可以不用从头计算最远点，而可以紧接着上一次的最远点继续计算。该算法的时间复杂度为 O(n)。因此整个算法的时间复杂度为 O(n)。



图一：旋转卡壳法

3 给定测度空间中位于同一平面的 n 个点，已知任意两点之间的距离 d_{ij}，存储在矩阵 D 中，求这组点的直径。该问题的直观解法就是把 D 扫描一遍，选择其中最大的元素即可。由于是在一个测度空间中，因此 d_{ij} 满足距离的基本要求，即非负性、对称性和三角不等式，我们就可以给出一种时间亚线性的近似算法。算法很简单，由原来确定性算法的检查整个矩阵改为只随机检查 D 的某一行，这样时间复杂性就由原来的 $O(n^2)$减少为 O(n)，相对于输入规模 n^2 而言，这是一个时间亚线性的算法。证明时间代价减小的同时，解不会小于最优值的一半

证明：
假设最大距离为 D_{ij}，随机选择了第 m 行，这一行最大的距离为 D_{mn}。那么：
$$D_{ij} < D_{im} + D_{mj} \text{（三角不等式）}$$
$$= D_{mi} + D_{mj} < D_{mn} + D_{mn} \quad \text{（对称性）}$$
$$\therefore D_{mn} > \frac{1}{2} D_{ij}$$
故时间代价减少的同时解不会小于最优值的一半。

0427

1 在平面上给定一个有 n 个点的集合 S，求 S 的极大点。极大点的定义：设 p1=(x1,y1)和 p2=(x2,y2)是平面上的两个点，如果 x1≤x2 并且 y1≤y2，则称 p2 支配 p1，记为$p_1 \prec p_2$。点集 S 中的点 p 为极大点，意味着在 S 中找不到一个点 q，q≠p 并且$p \prec q$，即 p 不被 S 中其它点支配。

不唯一

A(3,4) B(1,2) C(1,5)

极大点：A、C

扫描线算法，从右往左移动。

找到 x_max后，之后就找 y_max

算法如下：

1） 找到 n 个点中 x 最大的点，时间为 O(n)

2） 判断这个点的 y 是否为最大的，是则该点为极大点，不是则这个点集没有极大点。时间为 O(n)

Px 比左侧大

Py 比右侧大

P 为极大点。

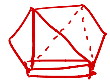
故这个算法的时间复杂度为 O(n)。

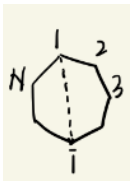
2 对凸多边形，1）有多少种三角划分的方法？2）如何使对角线长度之和最小？

如果觉得递推式不好求，写出递推式就可以了

1) $C(3)=1$ $C(4)=2$ 　从边

$C(n) = \sum_{k=3}^{n-2} C(k)C(n-k+1) + 2C(n-1)$





1） 对于要分割的多边形，我们从节点 1 开始考虑：

它只能与 3,4，···，N-1 相连，当它与节点 i 相连时，会分割出凸多边形 i 和 N+2-i。

故由节点 1 划分的方法有 E_1（N）＝E（3） E(N+2-3)+E(4)E(N+2-4)+ ··· +E(N-1)E(3)

其余每个点类似，由于每条边会被重复计算两次，故总共有：

$$E(N) = \sum_{i=3}^{N-1} f(i)f(N+2-i)\cancel{\frac{N}{2}}\ \frac{n}{2(n-2)}$$ 种方法。

2） 首先选择一条最短的对角线，该对角线将多边形分为边数较少的两个多边形，之后再分别选择两个多边形里最短的对角线，重复这样的操作直到完全划分。计算选出的对角线之和。

设 $i<j$，则由顶点 $i, i+1, \ldots, j$ 构成的凸多边形添加对角线的长度之和最小值为 $d[i,j]$

则 $d[i,j] = \min_{i+1\le k\le j-1}\{d[i,k]+d[k,j]+l_{kj}+l_{ki}\}$，$d[i,i]=0$

3 给定平面上 n 条线段，设计算法用 O(nlogn)时间确定其中是否有两条线段相交。

算法如下：

利用扫描线算法，事件点为线段端点及线段交点，并按点的 x 坐标排序，依次在这些点进行处理：

- 线段 s 为垂直线段：检查该线段与在 S 中的所有线段是否有交点，不将 s 加入 S 中。
- 线段 s 的左端点：
  1） 将线段加入动态集合 S 中
  2） 检查 s 在 S 中的相邻线段与 s 是否有交点

课上算法.

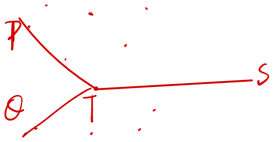
- 线段 s 的右端点：
  1） 从动态集合中删除线段
  2） 检查 s 在 S 中的相邻线段之间是否有交点

$(x,y) = \sum \alpha_i (a_i,b_i)$ $(0\le\alpha_i\le 1)$

4 有 n 种液体 $S_1,S_2,\cdots,S_n$，都含有 A,B 两种成分，含量分别为$\{a_1,b_1\},\{a_2,b_2\},\cdots,\{a_n,b_n\}$，$a_i+b_i<100\%$。现欲利用这 n 种液体配制目标液体 T，使之 A 和 B 的含量分别为 x 和 y。设计算法判别能否成功配制，并给出算法时间复杂性。

法一：判别 (x,y) 是否在 $\{a_1,b_1\},\{a_2,b_2\},\ldots\{a_n,b_n\}$ 构成的凸包中，时间 O(nlogn)

法二：以 T(x,y) 为原点



$\angle PTQ < 180^\circ$ 内部

$\angle PTQ > 180^\circ$ 外部.

线性 O(n)

该问题与 01 背包问题相似，可以首先求出配置液体使 A 的含量为 x 的配置所有方案，之后再判断这些方案是否可以使得 B 的含量为 y。
假设 F[i][j]表示前 i 种液体能否使 A 含量等于 x，如果不等，则值为一个无穷小的数，如果相等，则值为 B 的含量。
动态转移方程为：$F[i][j] = \max\{F[i][j-1], F[i-1][j-a_i]+bi\}; F[0][0]=0;$
最后判断 A 的含量等于 x 的所有结果中是否有 B 的含量等于 y 的即可。

算法复杂性：O（xn）