计算几何选讲

董炜隽

广州市第六中学

2018 年 3 月 29 日

点积

$a \cdot b = |a||b| \cos\theta$，其中 θ 表示向量 a 旋转到向量 b 所经过的夹角。

其绝对值等于向量 a 在向量 b 上的投影的模长乘上向量 b 的模长。

设 $a = (x_1, y_1), b = (x_2, y_2)$，则它们的点积等于 $x_1 x_2 + y_1 y_2$。

简单应用：两个向量垂直 $\Leftrightarrow$ 其点积为 0。

叉积

$a \times b = |a||b| \sin\theta$，其中 θ 表示向量 a 旋转到向量 b 所经过的夹角。

几何意义为，以两个向量为邻边构成的平行四边形的有向面积。当 b 在 a 左侧时为正，右侧时为负。

设 $a = (x_1, y_1), b = (x_2, y_2)$，则它们的叉积等于 $x_1 y_2 - x_2 y_1$。

简单应用：两个向量平行 $\Leftrightarrow$ 其叉积为 0。

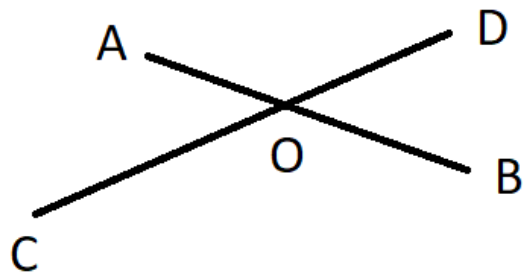
旋转向量

把向量 $a=(x,y)$ 旋转 θ 弧度后得到的坐标为

$$(x\cos\theta-y\sin\theta, x\sin\theta+y\cos\theta)$$

证明：考虑先把 a 放缩成一个模长为 1 的向量，设其与 x 轴的夹角为 α，那么显然 $x=|a|\cos\alpha, y=|a|\sin\alpha$。旋转后其实就是要求 $\cos(\alpha+\theta)$ 和 $\sin(\alpha+\theta)$，直接套用和角公式即可。

直线的交点



先使用叉积求出 $\triangle ABC$ 和 $\triangle ABD$ 的面积，得到 OC 和 OD 的比值，然后就可以求出 O 点的坐标了。

多边形面积

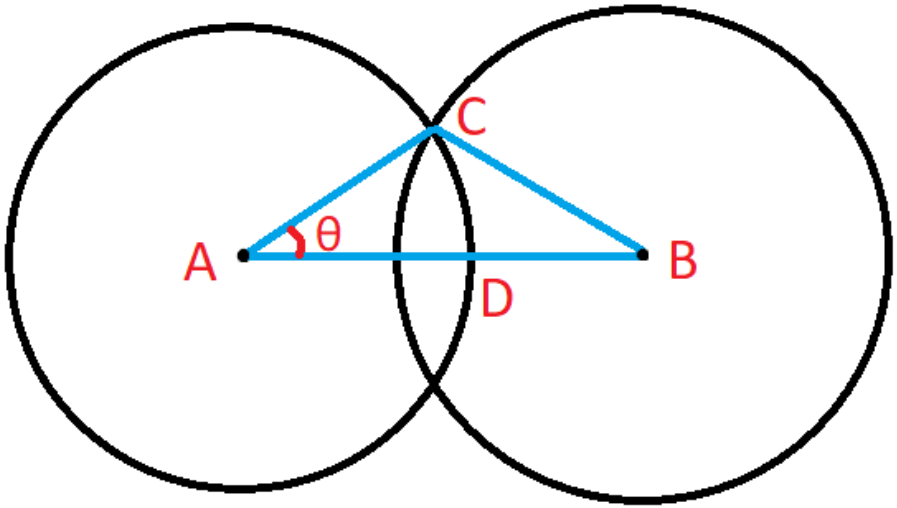
已知一个简单多边形的顶点按逆时针顺序依次为 $P_1, P_2, ..., P_n$，其面积为

$$\frac{1}{2}\left(P_n \times P_1 + \sum_{i=1}^{n-1} P_i \times P_{i+1}\right)$$

为什么是对的？

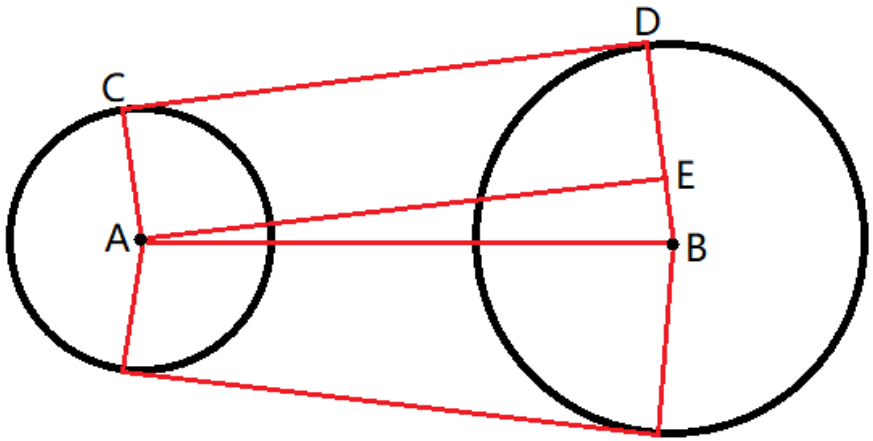
考虑平面上的一个点，我们以它为起点引一条反向延长线过原点的射线。如果该点在多边形外，这条射线会经过偶数条多边形的边，最终这个点的面积不会被算入答案中；如果在多边形内则会经过奇数条边，相抵消后恰好计算了一次它的面积。

圆的交点



用余弦定理算出 θ，然后把 D 点绕圆心 A 旋转即可求出交点 C 的坐标。

两圆公切线



用 $|AB|$ 和 $|BE|$ 算出 $\angle BAE$，剩下的就很好做了。

内公切线的情况也类似。

凸包

一个点集的凸包就是能包围给定点的最小的凸多边形。

按极角序/水平序排序后使用单调栈维护。复杂度的瓶颈在排序上。

一些与凸包相关的询问

- 查询一个点是否在凸包内？
  按照极角序/水平序二分出在哪两个点之间，判一下是否在该线段内侧。

一些与凸包相关的询问

- 查询一个点是否在凸包内？
  按照极角序/水平序二分出在哪两个点之间，判一下是否在该线段内侧。
- 一条斜率为 k 的直线从正无穷远处向下平移，问碰到的第一个点？
  在凸包上二分斜率。

动态凸包

- 只有插入，离线：CDQ 分治。$O(n \log n)$
- 只有插入，在线：用平衡树维护。$O(n \log n)$（一个小技巧：把坐标取负之后再维护一个下凸包就不用写两遍了）
- 要求插入和删除，离线：把每个点视为在时间 $[l, r]$ 内有效，然后外面套一个线段树分治。$O(n \log^2 n)$
- 要求插入和删除，在线：平衡树套可持久化平衡树维护，详见《可持久化数据结构研究》。$O(n \log^2 n)$

怎么造一个凸多边形

随机若干个和为零的向量，按极角排序后首尾相接即可。注意所有向量横纵坐标的绝对值加起来不能超过数据范围。

整点凸包的点数上界

假设所有点都是整点，并且坐标的范围为 n，那么凸包上的点数是 $O(n^{\frac{2}{3}})$ 的。

我们只考虑右下凸包，其余部分同理。只需证明分子分母和最小的前 $O(n^{\frac{2}{3}})$ 个既约分数的分子分母和能达到 $O(n)$ 级别。分子分母和为 i 的既约分数显然不超过 i 个，那么分子分母和前 k 小的既约分数的分子分母和至少也是 $k^{1.5}$ 级别的。

Codeforces 605C Freelancer's Dreams

有 n 种工作，做 1 单位时间的第 i 种工作可以获得 a_i 经验和 b_i 金钱，现在需要 p 经验和 q 金钱，问至少需要工作多久。工作时间可以不是整数。$n \leq 10^5$。

Codeforces 605C Freelancer's Dreams

有 n 种工作，做 1 单位时间的第 i 种工作可以获得 a_i 经验和 b_i 金钱，现在需要 p 经验和 q 金钱，问至少需要工作多久。工作时间可以不是整数。$n \leq 10^5$。

相当于有 n 个向量 (a_i, b_i)，平均每单位时间的收益就是这些向量的线性组合。考虑把这些点和 $(0,0), (maxa, 0), (0, maxb)$ 一起求个凸包，那么凸包内的点就是所有可能的单位时间收益。所以把从原点到 (p, q) 的射线和凸包求个交即可。

清华集训 2011 集合的面积

给出两个点集 A, B，定义点集 $A + B = \{a + b | a \in A, b \in B\}$，求点集 $A + B$ 的凸包的面积。

$|A|, |B| \leq 10^5$，坐标的绝对值不超过 10^8。

清华集训 2011 集合的面积

先分别求出点集 A 和点集 B 的凸包。记 a_i, b_i 分别表示 A 或 B 的凸包上的第 i 个点。

首先显然 $a_1 + b_1$ 在 $A + B$ 的凸包上。假设我们已知 $a_i + b_j$ 在 $A + B$ 的凸包上，可以发现凸包上的下一个点就是 $a_{i+1} + b_j, a_i + b_{j+1}$ 中更凸的一个。因此拿两个指针用类似归并排序的方式扫一遍即可。

清华集训 2011 集合的面积

- 给定两个点集，问点集之间的最远点距离。

清华集训 2011 集合的面积

- 给定两个点集，问点集之间的最远点距离。
- 给三个凸多边形，在三个凸多边形内部各选一个点，每次询问这三个点的重心是否可能是某个点。

对踵点

对于凸多边形上的两个点，如果能过它们画两条平行直线，且整个凸多边形都位于两条直线之间，我们就称它们为一对对踵点。

我们可以使用旋转卡壳算法求出凸多边形的所有对踵点。

旋转卡壳

平行卡壳分为两种情况，一种是卡住两个点，另一种是卡住了至少一条边。实际上我们只需要维护第二种情况就能求出所有的对踵点了。

我们顺次枚举凸包上的每条边，当它被平行卡壳卡住时，另一条直线一定是卡住了离这条边最远的点。我们找到这个最远点后，它与当前这条边的两个端点都能构成对踵点。

显然凸包上的点到固定边的距离是单峰的。因此我们枚举下一条边的时候只需要不断把当前最远点往后移动即可，并且最远点移动的过程中经过的点都能与这条边的端点构成对踵点。

另外，如果当前边的最远点在一条平行边上的话，两条边之间的四对点都可以成为对踵点。

旋转卡壳

旋转卡壳的过程中最远点不会旋转超过一圈，因此整个过程的复杂度以及对踵点的对数都是线性的。

旋转卡壳的应用

- 凸多边形直径
- 凸多边形间最大/最小距离
- 凸多边形最大面积/周长外接矩形

集训队互测 2012 最远点

给一个 n 个点的凸多边形，求距离每个点最远的点。

$n \leq 5 * 10^5$。

集训队互测 2012 最远点

给一个 n 个点的凸多边形，求距离每个点最远的点。$n \leq 5 * 10^5$。

需要注意的是旋转卡壳并不能求每个点的最远点。

集训队互测 2012 最远点

给一个 n 个点的凸多边形，求距离每个点最远的点。$n \leq 5 * 10^5$。

需要注意的是旋转卡壳并不能求每个点的最远点。

对于任意的两个点，我们作出它们的垂直平分线，那么在这条线的两侧分别是这两个点更优。因此这题是具有决策单调性的。但是环要怎么处理呢？可以直接倍长一下，然后规定第 i 个点的决策点只能在 $[i+1, i+n]$ 内选就好了。

半平面交

半平面是指二维平面上一条直线的一侧的所有空间。通常使用点对 (P, Q) 表示在 $\overrightarrow{PQ}$ 左侧的半平面。

多个半平面的交要么是空集，要么是非空凸集。

常用的做法是按极角排序后用双端队列维护，复杂度的瓶颈同样在排序上。

凸多边形求交

不妨无脑上半平面交，排序可以直接归并，复杂度是线性的。

UOJ242 破坏蛋糕

平面上有 $n+1$ 条直线，前 n 条直线把平面分成许多块，这些块有些面积有限，有些面积无限，而第 $n+1$ 条直线不经过前 n 条直线的交点，且一定不和前 n 条直线中的任意一条平行，求第 $n+1$ 条直线被前 n 条直线划分成的 $n+1$ 段中哪些在面积有限的块里，哪些在面积无限的块里。$n \leq 10^5$。

UOJ242 破坏蛋糕

考虑对于每一段所在的区域用半平面交判断是否有界。可以在预处理时先按极角排好序。每次走到下一段时只会有一个半平面的方向反了过来，直接插回去就可以了。时间复杂度 $O(n^2)$。

有没有更简洁的判定方法呢？实际上我们只需要判断是否有相邻的两个半平面转过的夹角大于等于 π，是的话就说明出现了一个开口。那么可以直接用一个 set 维护半平面的极角，每次修改时查一下前驱后继就好了。时间复杂度 $O(n \log n)$。

最小圆覆盖

求一个最小的圆，使得给定的所有点在圆的内部（或者边界上）。

通常使用经典的随机增量法。

随机增量法

- 把点的顺序随机打乱后依次加入，加入点 i 时，若点 i 不在前 $i-1$ 个点的最小覆盖圆内，说明它在新圆的边界上，那么变成下面的子问题。
- 已知点 i 在圆的边界上，求前 i 个点的最小覆盖圆。依次加入每个点，加到点 j 时如果 j 不在前 $j-1$ 个点和点 i 的最小覆盖圆内，说明它在新圆的边界上，同样会变成一个子问题。
- 已知点 i,j 在圆的边界上，求前 j 个点和点 i 的最小覆盖圆。依次加入每个点，加到点 k 时如果 k 不在前 $k-1$ 个点和点 i 及点 j 的最小覆盖圆内，说明最小覆盖圆上的三个点就是 i,j,k 了。

随机增量法

来分析一下时间复杂度。

最底层的子问题复杂度显然是 $O(n)$ 的。

对于第二层的子问题，当枚举到 j 时会有 $\frac{3}{j}$ 的概率需要做一遍 $O(j)$ 的下一个子问题，因此复杂度是 $O(n)$ 的。

同理可得第一层也是 $O(n)$ 的。这样我们就得到了一个优秀的线性做法。

扫描线

扫描线也是计算几何中常用的方法。核心思想是把二维平面通过某种方式切成若干个区域，使得图形的相对位置在每个区域内部不发生改变，然后对每个区域分别进行计算。

扫描线

- 矩形面积并

  相信大家都会。

- 求若干条线段之间的所有交点

  把线段的端点和所有交点的横坐标当做事件点做扫描线。由于交点是动态产生的，需要用一个堆来维护最靠前的一个交点。时间复杂度 $O((n+k)\log n)$，其中 k 为交点个数。

平面图点定位

给若干个询问点，问它们分别在平面图的哪个域中。

平面图点定位

给若干个询问点，问它们分别在平面图的哪个域中。

先来讲一下怎么求出平面图的所有的域。

- 把每条无向边拆成两条有向边，之后把每个点的出边按极角排序。
- 从任意一条未访问的有向边出发，在当前边终点处选择该边反向边的顺时针方向下一条边接着走。走回起点时就找到了一个新的域。
- 不断重复上面的过程直到走过了所有的边。

无界的域可能需要特殊处理，可以用叉积算有向面积来判断，如果不是正的就说明是无界的。

平面图点定位

给若干个询问点，问它们分别在平面图的哪个域中。

先来讲一下怎么求出平面图的所有的域。

- 把每条无向边拆成两条有向边，之后把每个点的出边按极角排序。
- 从任意一条未访问的有向边出发，在当前边终点处选择该边反向边的顺时针方向下一条边接着走。走回起点时就找到了一个新的域。
- 不断重复上面的过程直到走过了所有的边。

无界的域可能需要特殊处理，可以用叉积算有向面积来判断，如果不是正的就说明是无界的。

求出所有域之后剩下的部分就很简单了。把所有端点和询问点拿出来做扫描线即可。

清华集训 2016 定向越野

二维平面上有 n 个互不相交的圆形水域，问从起点走到终点并且不穿过任何圆形水域的最短路线是多长。$n \leq 500$。

清华集训 2016 定向越野

二维平面上有 n 个互不相交的圆形水域，问从起点走到终点并且不穿过任何圆形水域的最短路线是多长。$n \leq 500$。

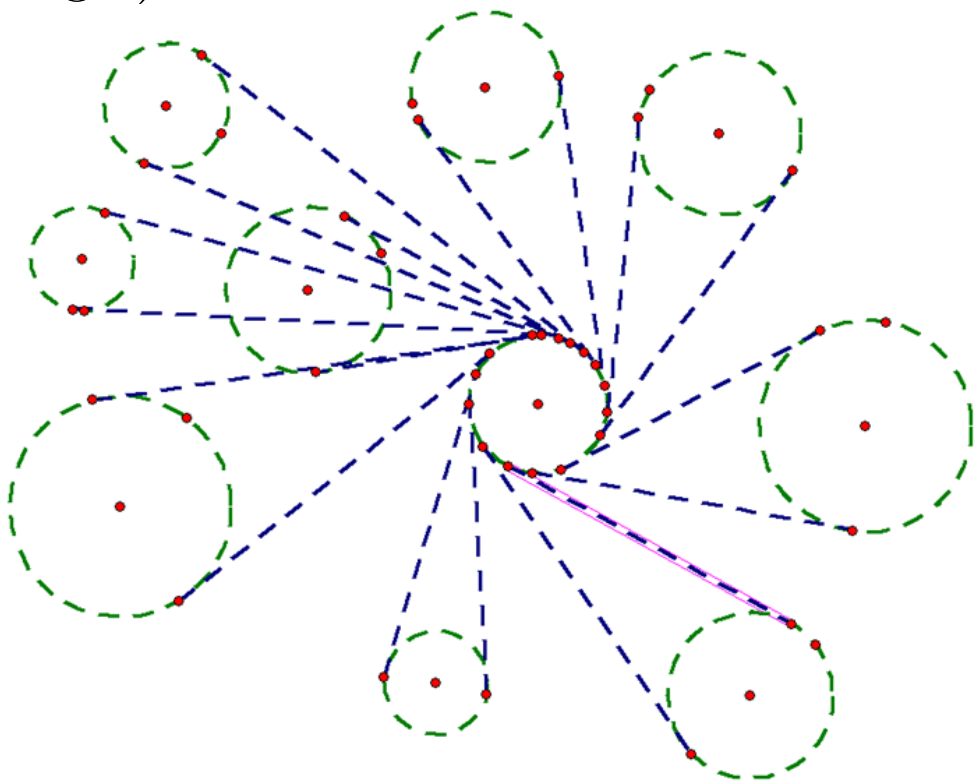
为了方便处理可以先把起点和终点看成半径为 0 的圆。我们可以求出任意两个圆之间的公切线，那么可以发现，最优方案的过程中一定只会沿着公切线走，或者沿着圆的边界从一个切点走到下一个切点。如果我们能求出每条切线是否被其他圆挡住，剩下的就只需要跑一个最短路了。

直接暴力判的话复杂度是 $O(n^3)$ 的，常数比较优秀的话可以通过。

清华集训 2016 定向越野

更好的做法？

考虑对于每个圆，把它与其他圆的所有顺时针/逆时针的公切线拿出来极角排序，然后用扫描线处理。时间复杂度 $O(n^2 \log n)$。



BZOJ 省选十连测 2017 Bridge

有一片 1000×1000 的平原，中间的一条河流把平原分成东西两部分，东岸和西岸都是两个端点分别在南北边界的折线。东岸有一个人位于 S 点，西岸有一个人位于 T 点，他们打算在河上修一条桥。假设桥分别修建在东岸的 A 点和西岸的 B 点，那么他们会分别从 S 走到 A、从 T 走到 B，然后在桥上见面。要求在桥最短的前提下最小化两人走的路程之和。

河流两岸的折点数目都不超过 20。

BZOJ 省选十连测 2017 Bridge

先考虑桥的两端至少有一个修建在顶点上的情况。显然 S 到 A 的路径一定是一条折线，并且折线上每个点都是输入给的点。那么把初始时直线可达的点对连上边然后跑最短路即可。

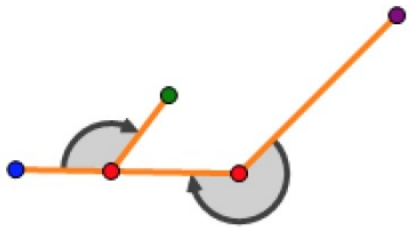
当桥的两端都不在顶点上时，说明两端所在的线段一定是平行的。如果两条线段之间的投影没有交集的话 A, B 一定就是最近的那对顶点了，已经在之前被考虑过。如果投影有交集，考虑枚举 S 到 A 路径上经过的倒数第二个点 P，以及 T 到 B 路径上经过的倒数第二个点 Q。A, B 连线一定是垂直于所在线段的，因此可以把线段平移一下，变成在线段上取一个点使得到 P, Q 的距离和最小的经典问题来解决。注意求出实际的 A, B 后还要判断是否合法。

清华集训 2017 我的生命已如风中残烛

墙上有 n 个钉子，第 i 个钉子的坐标是 (x_i, y_i)。还有 m 根绳子，绳子的一端是点 s_i，经过点 t_i 且长度为 L_i。其中 s_i 是粘在墙上的，而另一个端点是可以移动的。

接着对每根绳子进行一次游戏，第 i 次游戏中可怜会捏着第 i 根绳子的活动端点进行顺时针移动。不难发现绳子每时每刻都在以某一个位置为圆心作顺时针的圆周运动。初始情况下圆心是绳子的固定端点 s，但是在运动的过程中圆心可能会不断发生变化，问圆心切换的次数。

$n, m \leq 500$。



清华集训 2017 我的生命已如风中残烛

预处理时先以每个点为中心把其他点极角排序。查询时每次找出当前方向下顺时针的下一个点继续绕，如果走出环了就把绳子长度模一下环长再继续做。可以发现每次走一圈后绳子的长度至少会减少一半，因此这样做总共只会走 $O(n \log L)$ 次。

每次找下一个点时直接二分会多一个 log。实际上直接从当前方向的后继开始暴力往后找就可以了，每一圈找下一个点的复杂度加起来是 $O(n)$ 的。

总复杂度 $O(n^2 \log n + nm \log L)$。

Thanks.