

# P5939 [POI 1998 R3] 折线 — 题目讨论

Claude & zzy

May 19, 2026

Luogu P5939 [POI 1998 R3] 折线

## 第 1 步：吃透题意

**Claude**

先不复述题面，直接拿样例 1 摸一遍。

五个点：(2, 3), (3, 4), (4, 5), (1, 6), (12, 27)，标准答案是 **3**。

一条“平直折线”从左往右一笔画过去，每一段和  $x$  轴夹角在  $[-45^\circ, 45^\circ]$ 。

问：你能把这 5 个点划成 **3** 条平直折线吗？写出每条折线具体覆盖哪几个点，并说一句你判断“这两个点能不能串在同一条折线上”靠的是什么。

**zzy**

这个也没什么好问的，直接过吧。

## 第 2 步：复杂度预算 / 数据范围

Claude

题意你说过了，那进“想算法”前先把预算钉一下——这一步最容易栽。

数据范围： $1 \leq n \leq 30000$ ,  $0 \leq x_i, y_i \leq 30000$ ，时限按常规 1s 算。

问：你心里想往哪个复杂度量级走？ $O(n^2) \approx 9 \times 10^8$  这种能不能直接顶过去，还是必须奔着  $O(n \log n) / O(n)$ ？给个目标量级 + 一句理由。

zzy

时间复杂度预期应该是  $O(n \log n)$ 。

## 第 3 步：建模——两点能否同折线的条件

Claude

$O(n \log n)$  合理，预算钉死。 $n = 3 \times 10^4$  时  $O(n^2) \approx 9 \times 10^8$  在 1s 下是 TLE 边缘， $O(n \log n)$  是稳妥目标。

进算法第一件事：把题目翻译成纯代数。设两个格点  $P = (x_1, y_1)$ 、 $Q = (x_2, y_2)$ ，不妨  $x_1 \leq x_2$ 。

问：“能用一段满足角度约束（与  $x$  轴夹角  $\in [-45^\circ, 45^\circ]$ ）的线段，把  $Q$  接在  $P$  后面”——把这件事写成一个关于  $x_1, y_1, x_2, y_2$  的精确不等式。

zzy

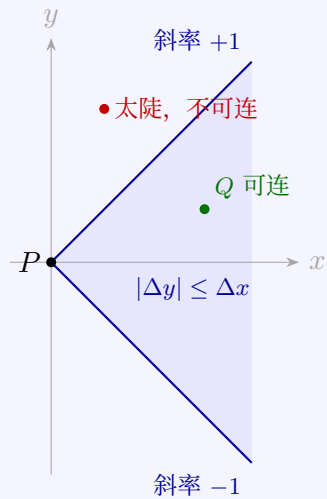
$x_2 - x_1, y_2 - y_1$  嘛，就是这个差的话，肯定是  $y_2 - y_1$  的绝对值，小于等于这个  $x_2 - x_1$ 。

Claude

对， $|y_2 - y_1| \leq x_2 - x_1$ （前提  $x_1 \leq x_2$ ）。夹角  $\leq 45^\circ \iff$  斜率绝对值  $\leq 1 \iff$  竖直跨度不超过水平跨度。

边界补一刀： $x_1 = x_2$  时退化成  $|\Delta y| \leq 0$  即  $P, Q$  同点——所以**两个不同的同  $x$  点永远进不了同一条折线**（左  $\rightarrow$  右折线对  $x$  严格单调）。这点记着，后面实现并列会用到。

几何上， $Q$  能接在  $P$  后面  $\iff Q$  落在  $P$  朝右张开的  $90^\circ$  锥形里（两条边界斜率  $\pm 1$ ）：



下一步只问一件：把这个带绝对值的不等式，拆成两条不带绝对值、同时成立的不等式。写出来就行。

zzy

$$y_2 - y_1 \leq x_2 - x_1 \text{ 且 } y_1 - y_2 \leq x_2 - x_1。$$

Claude

对，两条同时成立：

$$y_2 - y_1 \leq x_2 - x_1 \text{ 且 } y_1 - y_2 \leq x_2 - x_1。$$

现在做一个纯移项的动作：把每一条都整理成“只含  $P$  的某个组合量  $\leq$  只含  $Q$  的同一个组合量”——也就是把下标 1 的项全挪到一边、下标 2 的项全挪到另一边，让  $P$  和  $Q$  在不等号两侧彻底分离。

问：移完项后，这两条不等式分别长什么样？冒出来的那两个“组合量”（关于一个点的  $x, y$ ）各是什么？

zzy

我懂了，就是需要  $x - y$  严格上升，且  $x + y$  严格上升是吧。

Claude

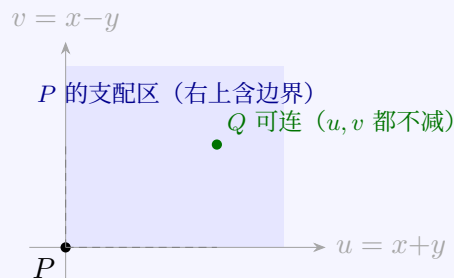
组合量抓对了——令

$$u = x + y, \quad v = x - y,$$

移项后两条恰好是  $u_P \leq u_Q$  且  $v_P \leq v_Q$ 。

一处要纠：是“**不下降** ( $\leq$ )”不是“严格上升”。原不等式就是  $\leq$ ，等号合法。反例： $P = (0,0)$ 、 $Q = (1,1)$ ， $u: 0 \rightarrow 2$  (升)， $v: 0 \rightarrow 0$  (不变，没严格升)；但  $|\Delta y| = 0 \leq \Delta x = 1$ ， $Q$  完全能接在  $P$  后面。若按“严格”就会把它误判成不能连。所以条件是**两维都不减** (dominance / 二维支配)，含等号。

把刚才那个“朝右  $90^\circ$  锥”换到  $(u, v)$  系里，锥的两条边正好贴着新坐标轴，“ $Q$  能接  $P$ ”就变成最干净的“ $Q$  在  $P$  的右上方 (含边)”：



## 第 4 步：换元后整道题变成什么

Claude

现在 pair 之间的关系干净了： $Q$  能紧接  $P \iff (u, v)$  两维都不减。把它从“一对点”抬到“一整条折线”、再抬到“整道题”——

问：一条平直折线覆盖的那串点 (按它经过的顺序)，在  $(u, v)$  坐标下构成什么形状 of 序列？于是“用最少的平直折线盖住全部  $n$  个点”这句话，等价地翻译成  $(u, v)$  点集上的什么问题？(用你自己的话说，先别管怎么算。)

zzy

(直接写了代码) 换元  $(x+y, x-y)$ 、按  $(u, v)$  排序、抽出  $v$  序列、 $\text{LIS}(uv, \text{false})$  输出长度。这个代码有什么问题？

## 第 5 步：代码 review —— Dilworth 哪个方向

Claude

你的代码说明 Step 4 你已经想通了：一条折线 =  $(u, v)$  两维都不减的链，“最少折线”= 最小链覆盖，并且想到用排序 + LIS 类做 Dilworth。框架对。

**Bug 在第 531 行** `LIS lis(uv, false);`——**Dilworth 方向反了**。

换元 + sort 后（你的 `Point::operator<` 恰好是  $u$  升、 $v$  升），一条折线 =  $v$  的一个非降子序列。“最少折线数”= 用最少数个“ $v$  非降子序列”盖住整串 = (Dilworth / 拦截导弹)  $v$  的最长严格下降子序列长度。而 `LIS(uv, false)` 算的是  $v$  的最长非降子序列 (`strict=false` 走 `upper_bound`)，量根本不对，只是样例 1 上巧合都 = 3 没暴露。

	代码算的	应该算的
<code>LIS(uv, false)</code>	$v$ 最长非降子序列	——
正确量	——	$v$ 最长严格下降子序列

反例（跑一下就现形）：输入 `2 / 0 0 / 2 0`。两点  $(0, 0), (2, 0)$  在同一条水平折线上（夹角  $0^\circ$ ），正解 = 1；代码： $(u, v) = (0, 0), (2, 2)$ ， $v$  序列  $[0, 2]$ ，最长非降 = 2 → 输出 2，错。

问：把它改成求  $v$  的最长严格下降子序列——你那个 LIS 类够用，想想 `strict` 参数 + 取负怎么配。改完用这个反例 + 两个样例自测，告诉我结果。