《大灾变》解析

目录

问题简述

有一座山脉，被描述为平面上的一条折线，折线可以被表述为一个函数的图像。要求(1)在山脉上建立一个高塔，使得站在塔顶可以看到整个山脉，且塔顶到塔底距离最小；(2)在山脉上空建立一个浮空岛，使得站在岛上可以看到整个山脉，且岛的海拔高度最小。

问题解析

思路甲

——欲窮千里目，更上一層樓[1]

看完这个问题，发现两问的共同点是都要找一个能够看见整个山脉的点，所以首先我们要解决的问题是，如何确定一个点是否可以看到整个山脉。根据定义显然是**从这个点到折线上任何一点的连线，只与折线交于这一点，或者与这一点所在的边重合（即与地面相切）**。由于“任意一点”有无穷多个，不便于我们直接判断，我们可以将定义转化为**从这个点可以看到折线上所有的线段（即所有的山坡）**，显然只有在线段所在直线上方才能看到整个线段，因此定义又等价于**这个点在折线上所有线段所在直线的上方**。

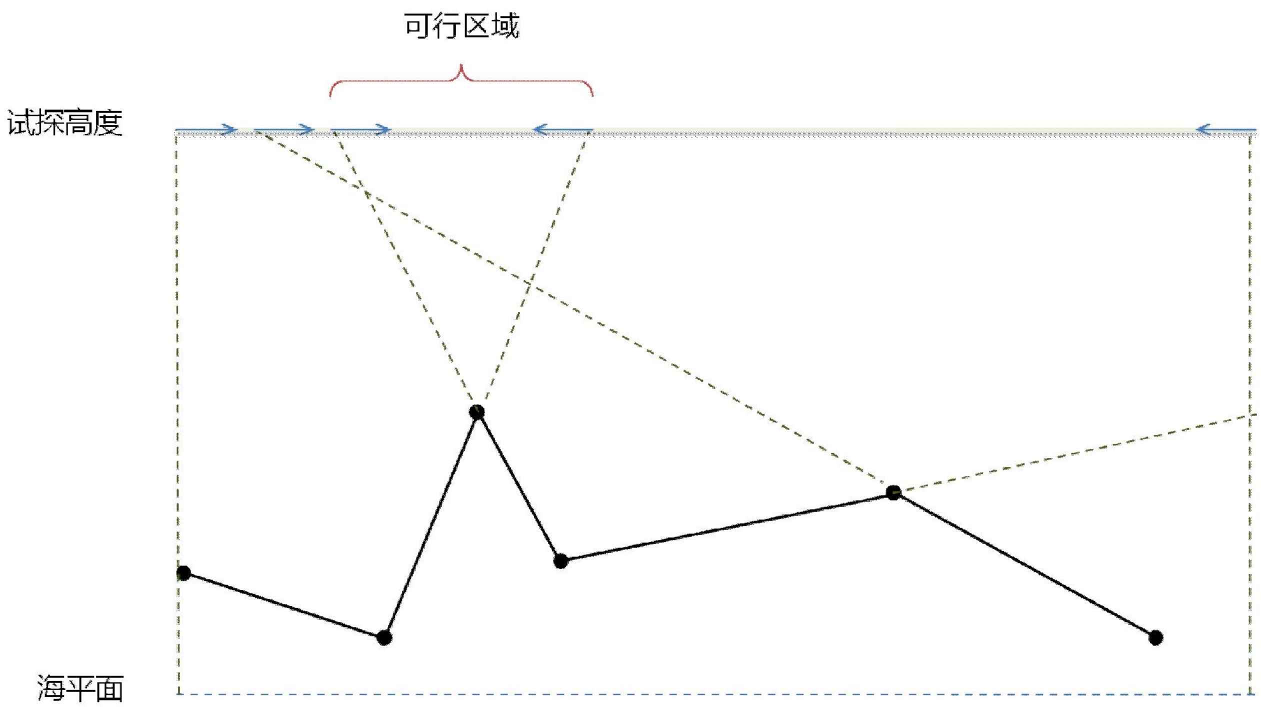
问题两问的不同点在于第一问为相对于下方山脉高度最低的点，第二问为相对于海平面高度最低的点。由于山脉情况复杂，似乎第二问比较容易求，因此我们从第二问开始考虑。从生活经验容易得知，站得越高，看得越远；站得越低，看得越近。我们容易想到：绝对高度升高，山脉可视部分不会减少；绝对高度降低，山脉可是部分不会增加。

- 定理 1
  - 绝对高度升高，山脉可视部分不会减少；绝对高度降低，山脉可是部分不会增加。
- 定理 1 证明
  - 山脉的可是部分可以定义为所有折线的可视部分的并集。假设当前的位置为(x0,y0)，高度变化量为Δy(>0)，升高后的位置为(x0,y0+Δy)。设当前位置可以看到某个山坡，即满足 y0>=k0*x0+b0。升高后的位置代入，代入得 y0+Δy>=k0*x0+b*，不等式仍然成立，所以该山坡仍可见。设当前位置不可以看到某个山坡，即满足 y0<k1*x0+b1。升高后的位置代入，若 Δy< k1*x0+b1-y0，则代入后 y0+Δy <k1*x0+b1 仍成立，即改山坡仍然不可见；若 Δy>= k1*x0+b1-y0，则 y0+Δy >=k1*x0+b1 成立，即山坡变为可见。因此绝对高度升高以后，可视范围不会减少。同理绝对高度降低以后，可视范围不会增加。

[1] 唐 · 王之涣 · 《登鸛雀樓》。

根据定理 1，我们有了优美的单调性，即如果站在一个点看山脉有看不见的地方，那就继续升高高度，如果可以看见整个山脉，那就降低高度。这样逐步缩小范围，直到找到“刚好”可以看到整个山脉的临界高度，这样就是最小的绝对高度了。接下来的问题就是给定一个高度以后，如何判断此高度上是否有点可以看见整个山脉。方法如下：

给定一个试探高度，将所有山坡线延长并交于试探高度线一点，直线上方区域映射在试探高度线上，为一个区间，如果在此高度上有点可以看到这个山坡，那么此点一定在这个区间内。将这样的所有区间求交集，就是整个山脉的可视区域。如果交集为空，那么就表示该高度上不能看到整个山脉，否则可视区域的最左端点就是可行解。



据此我们可以得出算法一。

算法一

——行远必自邇，登高必自卑[2]

1. 确定高度范围，二分试探高度。
2. 对于确定高度，求所有山坡可视范围在试探高度线上映射的交集。
3. 若交集为空，则无解，升高试探高度；否则有解，降低试探高度。
4. 当高度范围缩小到 10^{-4} 一下，当前高度就是最小绝对高度，答案为当前高度上可行区间的最左端顶点。

[2] 戰國·《禮記·中庸》

算法甲分析

若最高可能的高度是 E，则二分高度时间复杂度为 O(logE)，对于每个确定高度，求可行区间的时间复杂度为 O(N)，因此时间复杂度为 O(NlogE)。

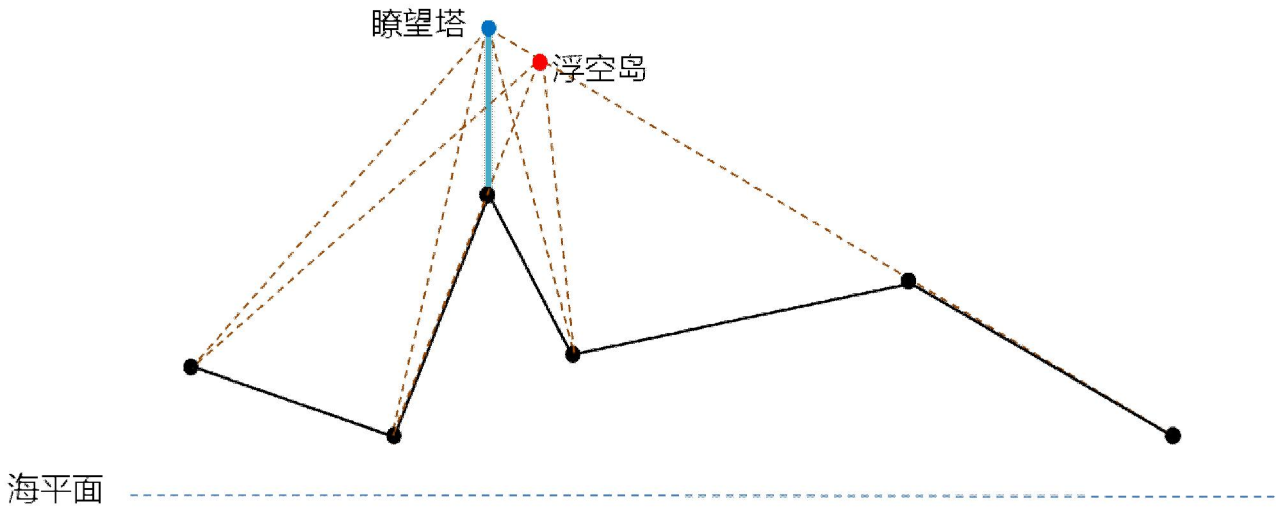
此方法只局限于求第二问（绝对高度最小），对于第一问（相对高度最小）暂时无法解决。

思路甲继续分析

虽然无法直接得出相对高度最低位置，但我们可以猜想相对高度最低位置与绝对高度最低位置的关系。

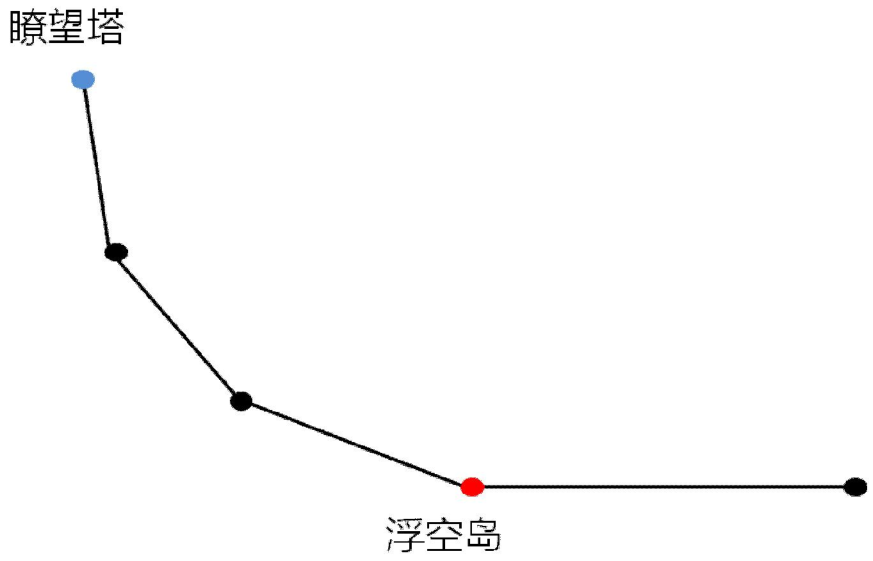
猜想 1 绝对高度最低位置就是相对高度最低位置

反例



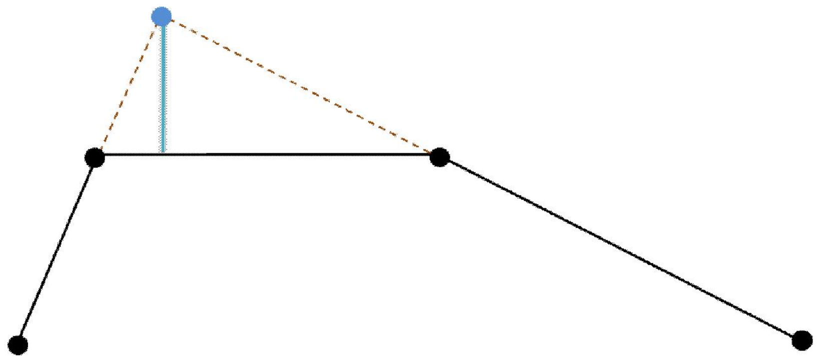
猜想 2 相对高度最低位置一定在绝对高度最低位置周围山坡上

反例



猜想 3 相对高度最低位置一定在山峰上

反例



看来第一问和第二问两者关系并非容易确定，我们只好改变思路。

思路乙

——不識廬山眞面目，衹緣身在此山中[3]

我们已经知道，折线上每条线段可以被看到，仅当在这条线段所在直线上方区域，即 $y>=kx+b$ 所确定的**半平面**，整个山脉的可视区域（下简称可视区域），就是所有**半平面的交集**。

- 定理 2
  - **可视区域的下边界一定是下凸的。**

[3] 宋 · 蘇軾 · 《題西林壁》

- 证明
  - (1)根据半平面交的性质，如果规定边界，那么半平面交的结果一定是一个凸多边形，因此规定上下左右边界后的可视区域是一个凸多边形。(2)因为每个半平面都是直线上方的部分（除上边界），所以可行区域内不存在一个点在任意一个直线下方，因此对于 x 轴上每一点都对应且只对应一定纵坐标在下边界上。根据(1)(2)，可视区域的下边界一定是下凸的。

根据定理 2，我们可以将可视区域描述为由一些一次函数（或常函数）构成的分段函数，设该函数为 f(x)。f(x)取得最小值时对应的点的位置，就是绝对高度最低点，即第二问的答案。由于 f(x)每段都是一次函数，而一次函数的最值在定义域端点上，所以 f(x)的最小值一定也在每段的端点上。扫描一遍 f(x)每个段的端点的函数值，即可求出最小值[4]。

由于山脉的 x 轴任何位置都只对应一个山脉的高度，山脉也可以表示为一个分段的一次函数[5]。设表示山脉折线的函数为 g(x)， h(x)=f(x)-g(x)则相对高度的最小值就是 h(x)最小值。

- 定理 3
  - 两个分段一次函数的差函数也是分段一次函数。
- 证明
  - 设 f(x),g(x)为两个定义域相同的分段一次函数，h(x)=f(x)-g(x)。对于 f(x)特定的一段，其定义域可能对应 g(x)的 k 段，对于对应 g(x)的 k 段中的特定一段，设 f(x) = k1*x+b1，g(x) = k2*+b2，h(x) = f(x)-g(x) = (k1-k2)*x + (b1-b2)，h(x)仍为一次函数（或常函数），因此对应 k 段分别为一次函数，h(x)的每一段均为一次函数。

所以当我们求出 h(x)后，可以仿照求 f(x)最小值的方法，求出 h(x)的最小值。

算法二

——表獨立兮山之上，云容容兮而在下[6]

1. 求所有山坡可视区域半平面的交集。
2. 求出表示山脉可视区域下边界的分段一次函数 f(x)。
3. 求 f(x)最小值，对应点就是绝对高度最低点（第二问“浮空岛”位置）。
4. 求出 f(x)与山脉函数 g(x)的差函数 h(x)，求 h(x)最小值，对应 x 在 f(x)上的点就是相对高度最低位点（第一问“瞭望塔”的位置）。

---

[4] 若有多个点均为最小值，去 x 最小者，下同。

[5] “分段一次函数”，指一个分段函数，每段都是一次函数（或常函数），下同。

[6] 戰國·屈原·《楚辭·山鬼》

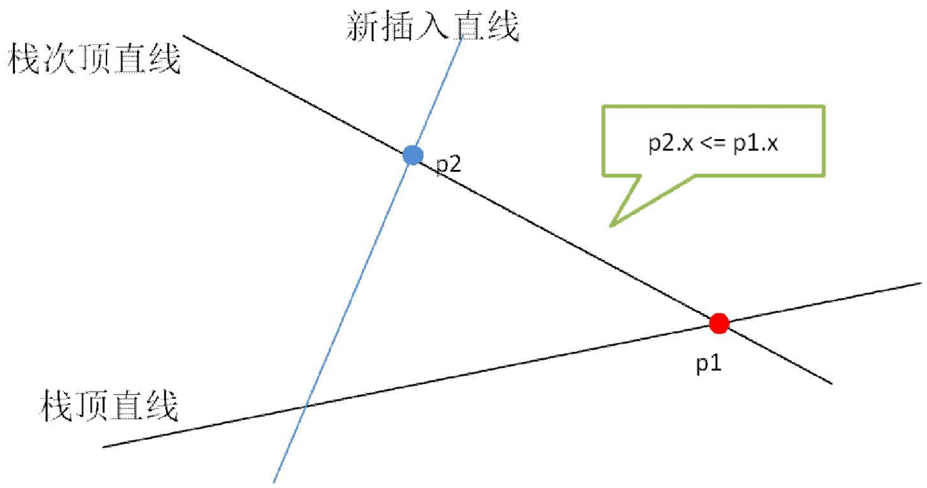
算法二分析

求两个分段一次函数的差函数，以及求分段一次函数的最值都是 O(N)的，算法的瓶颈在于求半平面交。已知的求半平面交的算法有 O(N^2)的朴素算法，和基于分制思想的 O(NlogN)的算法。不过该算法实现起来繁琐，不利于编码实现。

思路乙继续思考

我们发现，表示山坡可视区域的半平面，都是在直线的上方，也就是不等式 y>=kx+b 表示的平面区域。由于可视区域下边界是下凸，也就是说具备了斜率的单调性，这一点在算法二中并没有被充分利用。

如果将所有直线按斜率从小到大排序，依次处理每一条直线，每次只保留**可能在最上方的直线**，就可以快速求出结果。用一个堆栈（或称“单调队列”）来维护目前最优的直线，按斜率从小到大顺序插入每一条直线。当插入下一条直线时，判断堆栈顶两直线交点 p1 的横坐标，是否大于等于新插入直线与堆栈次顶[7]的直线交点 p2 的横坐标。如果是，则说明新插入直线比当前栈顶直线更优，删除栈顶直线，然后重新判断，直到将非最优的直线都删除，再将新的直线插入。



上图新插入直线较栈顶直线优，删除栈顶直线

算法三

——刪繁就簡三秋樹，領異標新二月花[8]

1. 将所有山坡直线按斜率排序排序。
2. 依次插入每条直线，用堆栈维护，保留最优的直线。
3. 根据堆栈中保留的最优直线，求出表示山脉可视区域下边界的分段一次函数 f(x)。

[7] 即堆栈顶下面的一个元素。

[8] 清 · 鄭燮 · 《題書齋聯》

4. 求 f(x)最小值，对应点就是绝对高度最低点（第二问“ 浮空岛 ”位置）。
5. 求出 f(x)与山脉函数 g(x)的差函数 h(x)，求 h(x)最小值，对应 x 在 f(x)上的点就是相对高度最低位点（第一问“ 瞭望塔 ”的位置）。

算法三分析

该算法成功得将求半平面交简化，使维护的时间变成 O(N)，加上斜率排序就是 O(NlogN)。该算法实现起来比算法乙要容易许多，算法时间常数也更小。

思路乙进一步思考

我们发现算法三中求山脉可视区域下凸线的主要瓶颈在于直线按斜率排序，如果能把这一步省去，算法将会变成线性[9]。如果还是用上述的方法，按从左到右的顺序插入直线，显然不能再用栈来维护，因为斜率不是单调的，我们必须“ 创造 ”其他的单调性。

一个直观的想法是，从左到右依次插入斜率递增的直线。这样的确可以求出一个下凸线，但可能不是合法的，因为略过了有些可能会决定最终高度的直线。相反，如果我们从右到左依次插入斜率递减的直线，也可能不合法。进一步考虑，如果把两者“ 结合起来 ”呢？经证明(见定理 4)这样结合起来的凸线就是合法的。两次维护的凸线，可以看做一个斜率序列中的两个单调上升子序列：一个从开头开始，一个在结尾结束。

- 定理 4
  - 可视区域下凸线上的直线，一定在斜率序列开头和结尾的两个单调子序列上。
- 证明
  - 上述陈述等价于“ 不在斜率序列开头和结尾的两个单调子序列上的直线，一定不是可视区域下凸线上的直线 ”（逆否命题）。假设存在一个不在两个子序列上的直线，它是可视区域下凸线上的直线，则它一定满足斜率大于凸线上左边一条直线的斜率，小于凸线上右边一条直线的斜率（下凸线的斜率单调性）。那么它也一定在开头或结尾两个字序列上，与假设矛盾，陈述得证。

因此，我们可以将改进算法三。

算法四

——近水樓臺先得月，向陽花木易逢春[10]

1. 从左到右依次插入斜率递增的直线，维护出凸线 A。
2. 从右到左依次插入斜率递减的直线，维护出凸线 B。
3. 求凸线 A、B 的交点，保留两者上方部分，合并为凸线 C。

---

[9] 除去最初的对山脉顶点排序，以下涉及线性者同理。

[10] 宋 · 俞文豹 · 《清夜录》

4. 凸线 C 就是求出表示山脉可视区域下边界的下凸线，表示为分段一次函数 f(x)。
5. 求 f(x)最小值，对应点就是绝对高度最低点（第二问“浮空岛”位置）。
6. 求出 f(x)与山脉函数 g(x)的差函数 h(x)，求 h(x)最小值，对应 x 在 f(x)上的点就是相对高度最低位点（第一问“瞭望塔”的位置）。

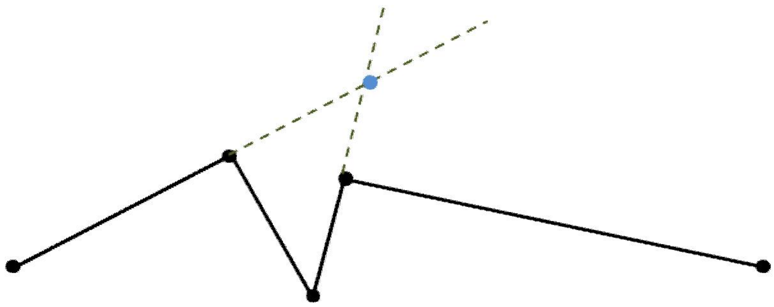
算法四分析

省去斜率排序的步骤，维护两个凸线，求其合并，我们成功地将算法优化到了线性。

思路丙

——會當淩絕頂，一覽衆山小[11]

虽然思路已的各种算法已经解决了这道题，但我们不妨还有别的思路。看到这道题以后，有一个直观的想法，猜想浮空岛（相对高度最低位置）就在所有的山坡直线交点的最高处。这样的反例是很容易举出的，例如：



观察发现这种反例只出现在两条上升直线[12]（或下降直线）在高处相交的时候，进一步猜想，是不是在所有的上升直线与所有下降直线相交位置中的最高点呢？除了全部是上升直线或者全部是下降直线（斜率大于 0），这个猜想是很难举出反例的。

事实上，这个算法是正确的，如何理解呢？我们把上升直线和下降直线分成两个集合 A 和 B 考虑（水平直线同时属于两个集合）。对于集合 A（上升直线）中直线表示的半平面，交集一定是下凸的（根据定理 2），同理集合 B 半平面交也是下凸的。设这两条下凸折线分别是 L_1 和 L_2，L_1 是每段斜率递增的下凸线，L_2 是每段斜率递减的下凸线。L_1 和 L_2 一定相交，且交于一个点 P（或斜率为 0 的边重合，认为交点是这条边上左端的那一点），那么“浮空岛”的位置一定在 P 上。因为实际上点 P 就是 L_1 和 L_2 表示半平面交的交集的最下方的点（反证：假设点 P 不是半平面交最下方点，则下方一定有一点 Q 是最下方点。若 Q 在 L_1 上，则一定有 Q 在 L_2 上方，则 Q 在 P 上方，矛盾，Q 在 L_2 上同理）。

[11] 唐 · 杜甫 · 《望嶽》

[12] 上升直线指的是斜率小于等于 0 的直线，下降直线是斜率大于 0 的直线，下同。

对于每一条上升直线，与其相交的所有下降直线的最高点，一定是下凸线 L_2 上（反证：如果不是的话，上方一定存在一条线比下凸线上该线优，则下凸线不是最优，矛盾）。同理对于每一条下降直线，与其相交的所有上升直线的最高点，一定在下凸线 L_1 上。因此所有上升和下降直线交点的最高点，一定既在 L_1 上，又在 L_2 上，即在 L_1 和 L_2 交点上。

算法五

——不畏浮雲遮望眼，自緣身在最高層[13]

1. 特判两种情况：全部是上升直线或全部是下降直线。
2. 对于每一条上升直线，求它与所有下降直线的交点，保留最高点。
3. 求出所有上升直线对应的最高点，保留最高点。
4. 最高点位置即为“浮空岛”位置。

算法五分析

求所有上升与下降直线交点的时间复杂度为 O(N^2)，只能求出“浮空岛”的位置。

位置关系的再思考

——此不爲遠者小而近者大乎[14]

随机生成一些数据，我们发现不少数据中，相对高度最低位置和绝对高度最低位置是同一点，而且山脉顶点数越多，随机范围越大，这一现象就越明显。为什么会有这种现象呢？其实我们可以这样考虑，当山脉地形越复杂，我们就要在越高的位置观察才能看到每一个细节。但实际上离得越远，就越难看清楚，即所谓“远者小而近者大”，就像在太空看地球，是难以看出山脉纵横的。也就是说，离得越远，地面就越趋近于平地，平地上相对高度最低位置和绝对高度最低位置一定是同一位置，因此会产生这种现象。

附录

灵感来源

- 居住在四川，看到地震灾区如火如荼的重建，感觉到人类战胜自然的伟大。
- 学习动态规划的单调性优化以后，一直在回味其思想，决定出一道与单调性有关的题。
- 在玩《魔兽世界》的时候，了解到资料片《大灾变》，看过介绍以后有感而发，写出了题面描述。

[13] 宋 · 王安石 · 《登飛來峰》

[14] 戰國 · 列禦寇 · 《列子 · 汤问》

试题考查点

- 计算几何基本知识
- 二分思想
- 半平面交
- 单调性的应用
- 分段一次函数求最值

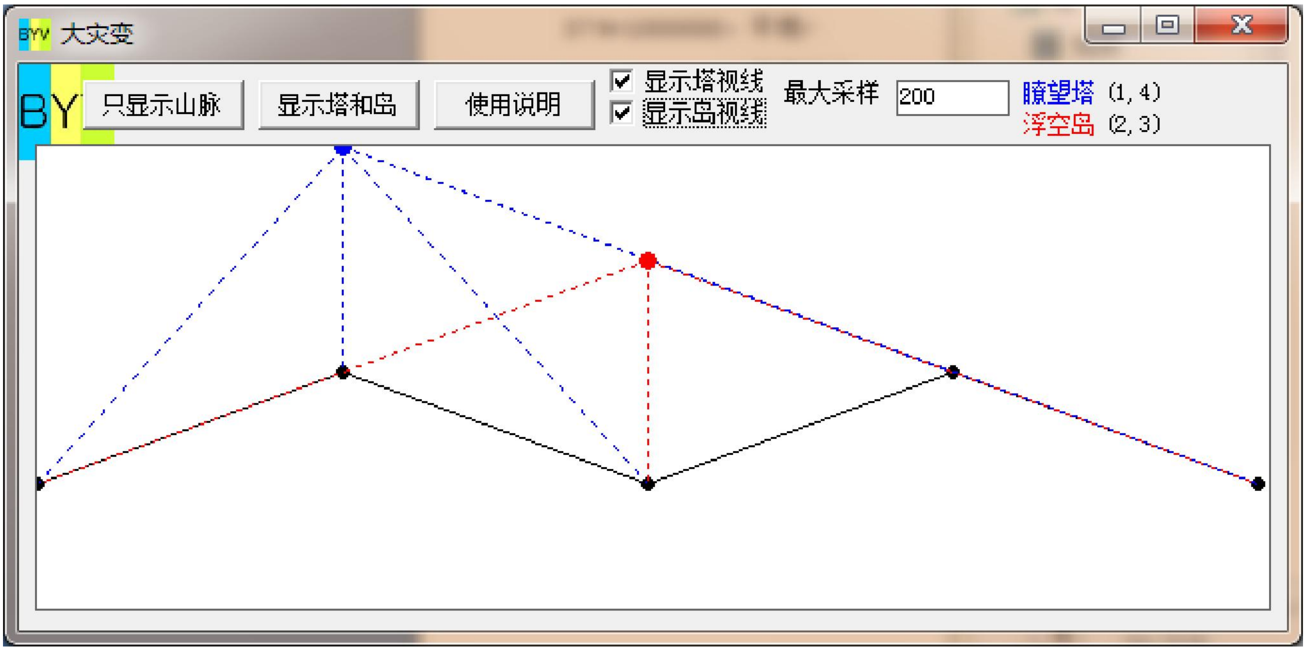
数据生成方法

01	有时候浮空岛可以在山谷地上，瞭望塔位置多解选横坐标最小
02	山脉是平地
03	两座山谷，两对并行线
04	视线范围超越山脉横坐标范围
05	一座山坡，极其陡峭，注意浮点误差
06	两座较高山峰，瞭望塔高度很高
07	平地上有个大裂缝，瞭望塔和浮空岛应该建立在裂缝边缘
08	瞭望塔建在山顶，浮空岛建在半山腰
09	两边高，中间低的盆地，注意防止浮点误差，避免-0.00
10	瞭望塔不在山脉顶点上
11 - 15	随机数据 N<=20，且瞭望塔不在山脉顶点上
16 - 20	随机数据 N<=1000
20 - 30	随机数据 N<=100000
31	有个深坑的下坡
32	平缓上坡
33	缓坡 + 深谷

34	深谷 + 缓坡
35	深谷 + 浅谷
36	近似一线的上坡
37	N=1000000，平地
38	N=1000000，高度差为 1
39	N=1000000，平地中有一个山峰
40	随机数据，N=250000，高度不超过 10

数据演示程序

使用方法，将 cataclysm.exe 和 calc.exe 放在同一文件夹内，打开 cataclysm.exe，把数据输入文件(.in)拖入界面方可显示数据。



估计分数

用算法一可以只求出第二问，至少能得到 50%的分数，由于不少数据第一问和第二问同解，所以期望得分为 65%。

N 大于 1000 的数据仅有 37.5%，因此使用算法二和算法五 O(N^2)的方法约能得到 62.5%的分数。

使用算法三和算法四可以得到 100%的得分，但由于有些细节容易被忽略，期望得分为 80%。

标题说明

思路甲	欲窮千里目，更上一層樓	说明“ 站得越高，看得越远 ”的道理
算法一	行遠必自邇，登高必自卑	太高了要降低高度，说明二分的原理
思路乙	不識廬山眞面目，祇緣身在此山中	说明可视区域在山脉上方
算法二	表獨立兮山之上，云容容兮而在下	在山脉上空的下凸线上扫描最低点，比喻为云层
算法三	删繁就簡三秋樹，領異標新二月花	用“ 删繁就简 ”概括维护凸线的堆栈（单调队列）
算法四	近水樓臺先得月，向陽花木易逢春	比喻制造单调性的优势
思路丙	會當淩絕頂，一覽衆山小	猜想到山坡交线最高点就能看见山脉
算法五	不畏浮雲遮望眼，自緣身在最高層	在上升线和下降线的最高点，就能一览山脉全貌