

# 卡常数

梁泽宇

Mato\_No1

CST, Tsinghua Univ.

# “卡常数”的来源

- 某日，某著名百度贴吧出现一个看似“正常”的帖子：

对于一道算法难题的疑惑

只看楼主

收藏

回复



楼主

\_叫我猪猪侠

普及一等



有几名别有用心的吧友指责我是在水贴，是在复制别人所说的语句。以牙还牙只能算作下策。

也有人质疑我这样做的目的何在。我本可以回答无可奉告，但我觉得有必要重申一遍，醉翁之意不在酒，我本意并非如此！我希望的只是唤醒吧友们的理性与良知，所以我感到有必要抛出一些学术问题，让noip吧重归正道。

下面这道难题已困扰我数年，我将它提出来，同诸位吧友们一起讨论，

http://codeforces.com/contest/314/problem/B。

# “卡常数”的来源

- jcvb 神犇很快给出了楼主所求助的题的解法——“卡常数”：



jcvb

提高一等



这是一道卡常数的好题！

# “卡常数”的来源

然而楼主却表示不知道什么是“卡常数”：



楼主

[\\_叫我猪猪侠](#)

普及一等



卡特兰数和括号序列之间的关系是众所周知的，我也曾试图将Catalan的理论应用到这道题中，但还没有得到令人满意的结果。

大部分具有特殊性质的数列都是由数学家的名字来命名的，比如卡特兰数、斐波那契数。但你们说的卡常数，我还真没听说过，在wikipedia上也找不到对应的资料。

或许是我孤陋寡闻，但我认为，在百度贴吧这个开放的平台下，更应该增进讨论，更应该把自己的观点详细无遗的展现出来。只留下一个深奥的名词就走人，令观者摸不着头脑，这样的回帖，绝对说不上是高质量的。

# “卡常数”的来源

- 那么“卡常数”到底是什么呢囧……
- 许多人提出了自己的见解：



成成8924

NOI铜牌



@叫我猪猪侠

我来给你讲讲什么叫卡常数

说古代有个人叫做卡

百度百科资料如下：

卡(Qa'a)是古埃及第一王朝的最后一位法老。其位于阿拜多斯的陵墓十分庞大，面积达到了98.5×75.5英尺(或30×23米)。据曼涅托之记载，卡的统治期长达26年——如果卡即为其书中记载的比厄彻尼斯(Biechenes)，而其位于阿拜多斯的庞大陵墓则支持了这一观点。20世纪年代中期，德国考古研究所的工作人员在卡的陵墓入口处发现了一枚刻有霍特普塞海姆威王名的印鉴——德国考古研究所在1993年挖掘了卡的陵墓，并在其四周发现了26个陪葬陵墓——这一发现证明了曼涅托的观点：即卡即埋葬于此，而其继任者为霍特普塞海姆威——后者是古埃及第二王朝的创建者。

这个卡发明了一个常数

叫做卡常数

懂了么

# “卡常数”的来源

- 那么“卡常数”到底是什么呢囧……
- 许多人提出了自己的见解：



GEOTCBRL

怒进省队



我是鱼

“卡常数”难道不是09年骆可强神犇提出来的么 😊 (感觉要卷入一番斗争了撒)

# “卡常数”的来源

- 那么“卡常数”到底是什么呢囧……
- 许多人提出了自己的见解:



lsm\_listen

提高二等



还有一个卡普雷卡尔(Kaprekar)常数(简称卡常数)。。。😊 ..... Refresh the lower limit of human knowledge

# “卡常数”的来源

- 后经查明，jcvb 神犇所说的“卡常数”是一种非常可怕的数列，由卡常（Karp-de-chant）发明，经过 Trajan 用 spaly 算法扩展后可以在线性事件内解决几乎一切 P 问题……
- 因发明、扩展了卡常数，卡常和 Trajan 分别被选为世界上平均智商最高的城市——智慧之城（City of Intelligence）的正、副市长……
- “人类智慧真是太可怕了”……



# 题意

- 三维空间中有  $N$  个摄像头，编号  $1 \sim N$ ，初始坐标已知；
- $M$  个操作，两种：
  - 修改某个摄像头的坐标；
  - 询问到某个点距离为给定值的摄像头编号（保证有且仅有一个摄像头满足要求）；
- 操作参数进行加密：
  - $f(x) = ax - b \sin x$ ；
  - 设上一次询问答案为  $last\_res$ ，之前无询问则  $last\_res = 0.1$ ，将参数加密成  $f(last\_res * \text{原值} + 1)$ 。

# 数据范围

测试点编号	N	M	附加信息
1	1024	1024	无
2	2048	2048	
3	32768	32768	没有修改操作
4	32768	65536	
5	65536	65536	
6	32768	32768	无
7	32768	65536	
8	65536	32768	
9	65536	65536	
10	65536	65536	

# 数据范围

- $0 \leq b < a < 5$  ;
- 所有坐标绝对值不超过 **100** , 至少精确到  $10^{-5}$  (数据中实际上精确到  $10^{-6}$ ) ;
- 参数加密后的值精确到  $10^{-15}$  (从 **sampIe** 中可以看出……) ;
- 坐标均为随机生成 (这一点很重要……呵呵……)

# 得分情况

- 70分：任之洲；
- 50分：张恒捷；
- 40分：王逸松、汪文潇、金策、于纪平；
- 30分：6人；
- 20分：14人；
- 10分：2人；

# 讨论时间

- 四……



# 如何解密

- 从  $f(x)=ax-bsinx$  中解出  $x$  ;
- $f(x)$  在实数域上严格递增……
- 证明:  $f'(x)=a-b\cos x$  , 由于  $a>b\geq 0$  , 所以  $f'(x)\geq a-b>0$  。
- 不会求导?? 可代入实际数值验证……
- 二分即可, 注意精度要足够高……

# 算法 0：暴力

- 对于修改操作，解密后直接修改；
- 对于询问操作，暴力枚举每个摄像头是否满足要求；
- 时间复杂度  $O(NM)$ ；
- 期望得分：20 ~ 30.....

# 算法 1：k-d 树

- k-d 树可以用来解决查找到一个点的距离为指定值的点的问题……
- 修改 = 删除 + 插入；
- 删除后不好维护？
- 给每个结点一个是否被删的标记，删除时直接将标记置为 **true**，遍历树的时候忽略已删除结点即可…  
…
- 由于点的坐标随机，树的深度不会很大，时间效率可以保证……
- 期望得分： 40 ~ 80 ；



, b

- 题目描述很逗……
- 标算更逗……
- 考验大家的人类智慧水平……

# 算法 e

- 对于操作中的每个参数，将其解密后可以知道  $\text{last\_ans} * \text{该参数原值}$ ，其中  $\text{last\_ans}$  为该操作上一次询问的答案；
- 假设我们已经知道了某个询问的答案为  $\text{No}$ ，以及此时编号为  $\text{No}$  的摄像头的坐标  $(x[\text{No}], y[\text{No}], z[\text{No}])$ ，则可得

$$\begin{aligned} & ((x_0/\text{last\_ans}) - x[\text{No}])^2 \\ & + ((y_0/\text{last\_ans}) - y[\text{No}])^2 \\ & + ((z_0/\text{last\_ans}) - z[\text{No}])^2 \\ & = (r_0/\text{last\_ans})^2 \end{aligned}$$

- 其中  $x_0$ 、 $y_0$ 、 $z_0$ 、 $r_0$  是已知的……

# 算法 e

- 这是一个关于  $(1/\text{last\_ans})$  的一元二次方程，由于  $\text{last\_ans} > 0$ ，所以可以乘  $\text{last\_ans}^2$  变为关于  $\text{last\_ans}$  的一元二次方程（可提高精度）；
- 其满足  $1 \leq x \leq N$  且为整数的根  $x$  即为  $\text{last\_ans}$ ；
- 若两根都是  $1 \sim N$  的整数，则枚举其一；若无  $1 \sim N$  的整数根，则说明该询问的答案有误——不是 No；
- 在没有修改操作的情况下， $(x[\text{No}], y[\text{No}], z[\text{No}])$  就是摄像机 No 的初始坐标……
- 枚举最后一个询问的答案，可以从后往前解出每个询问的答案；
- 需要搜索（可能两个根都是  $1 \sim N$  的整数），由于坐标随机，这种情况很难出现，所以搜索量不大……
- 期望得分 30，配合算法 0（暴力）可得 50 分……

# 算法 $e^e$ (标算)

- 有修改操作肿么办??
- 得到一个询问的答案  $No$  之后, 为了得到摄像头  $No$  目前的坐标, 需要找到上一次对  $No$  的坐标进行修改的操作……
- 注意到, 修改操作中的摄像头编号  $i$  也被加密了, 解密后得到  $last\_ans0 * i$  的原值, 设为  $i0$  (这里的  $last\_ans0$  为相对于该修改操作的  $last\_ans$ );
- 由于  $last\_ans0$  是整数, 所以对摄像头  $No$  的修改操作, 其  $i0$  值必然是  $No$  的倍数;
- 求每个修改操作的  $i0$  值的因数, 找到所有可能的  $i$  的原值……

# 算法 $e^e$ (标算)

- 处理某个答案为 **No** 的询问时，从后往前枚举所有可能修改摄像头 **No** 的操作，并将其代入验证（在确定某修改操作的 **i** 时，其 **x**、**y**、**z** 也同时确定）…  
…
- 优化：若某次询问的枚举中找出了某个修改操作不可能修改 **No**，则下次再出现答案为 **No** 的询问时，该修改操作可以跳过……
- 时间复杂度：  $O(M*\sqrt{N})$
- 期望得分： **100**……

# 总结 && 吐槽

- 一般题目使用通过 `last_ans` 加密的手段，都是为了强制在线……
- 但有时这可以给我们通过加密结果直接解出 `last_ans` 的机会……
- 这种思想最初出自人类智慧教主许昊然 (`clevertick`) 的《一道防 AK 好题》（具体见今年冬令营讲稿）……
- 后来被引入了 `SDOI` 的某题……

# 总结 && 吐槽

- 启示:
- 我们一定要**多想想**……
- 想不出，换个方向**再想想**……



pearfish16

省选酱油



想不出TT

举报 | 20楼 2013-06-28 14:12

收起回复



sillycross: 再想想

2013-6-28 07:58 回复

谢谢大家！！

我们要充分发挥人类智慧、探索、测试、改进解决问题的能力……