


A collection of historical artifacts is arranged on a light-colored surface. In the top left, a portion of a wooden chessboard with a checkered pattern and several chess pieces is visible. Below the chessboard, there are two medals: one with a red ribbon and a circular emblem, and another with a blue ribbon and a circular emblem. To the right of these medals is a silver star-shaped medal with a central emblem. In the bottom left corner, there is a round compass with a white face and black markings. A pair of round-rimmed glasses with thin metal frames lies diagonally across the lower half of the image.

# 猜数问题的研究

——《聪明的学生》一题的推广

上海市复旦附中  
张宁




近年来，信息学奥赛的试题涵盖面越来越广，不仅在程序设计方面对选手掌握算法与数据结构的要求越来越高，对选手的数学水平也提出更高的要求。

而数学类问题的难度，并不在于编程，而在于思想。其中也不乏一些比较另类的数学题，如 CTSC2001

《聪明的学生》这样一道逻辑推理问题与竞赛中常考的组合数学题目不同，它并不要求选手掌握任何高深的数学知识，但对选手的抽象思维能力提出了挑战。

我个人对这个有趣的问题比较感兴趣，对题目进行了深入的思考，并将其推广到一般情形。下面将主要从数学证明的角度来分析问题，由于时间关系，我将略过原问题的证明，以及第一种简单的推广，选择第二种推广的部分证明进行讲解。（论文中有详细证明）



- 
- ◆ 问题描述
  - ◆ 两个例子的分析
  - ◆ 问题分析
  - ◆ 一些定义
  - ◆ 思路分析
  - ◆ “终结情形”的条件
  - ◆ “一类情形”能否转化为“终结情形”
  - ◆ “二类情形”能否转化为“终结情形”
  - ◆ 编程实现中的若干问题
  - ◆ 结束语

## 问题描述：

一位逻辑学教授有  $n$  名非常善于推理且精于心算的学生。有一天，教授给他们  $n$  人出了一道题：教授在每个人脑门上贴了一张纸条并告诉他们，每个人的纸条上都写了一个大于 0 的整数，并将他们分成了两组（一组学生有  $m$  人， $m \geq n/2$ ，学生并不知道如何分组），两组学生头上数的和相等。于是，每个学生都能看见贴在另外  $n-1$  个同学头上的整数，但却看不见自己的数。

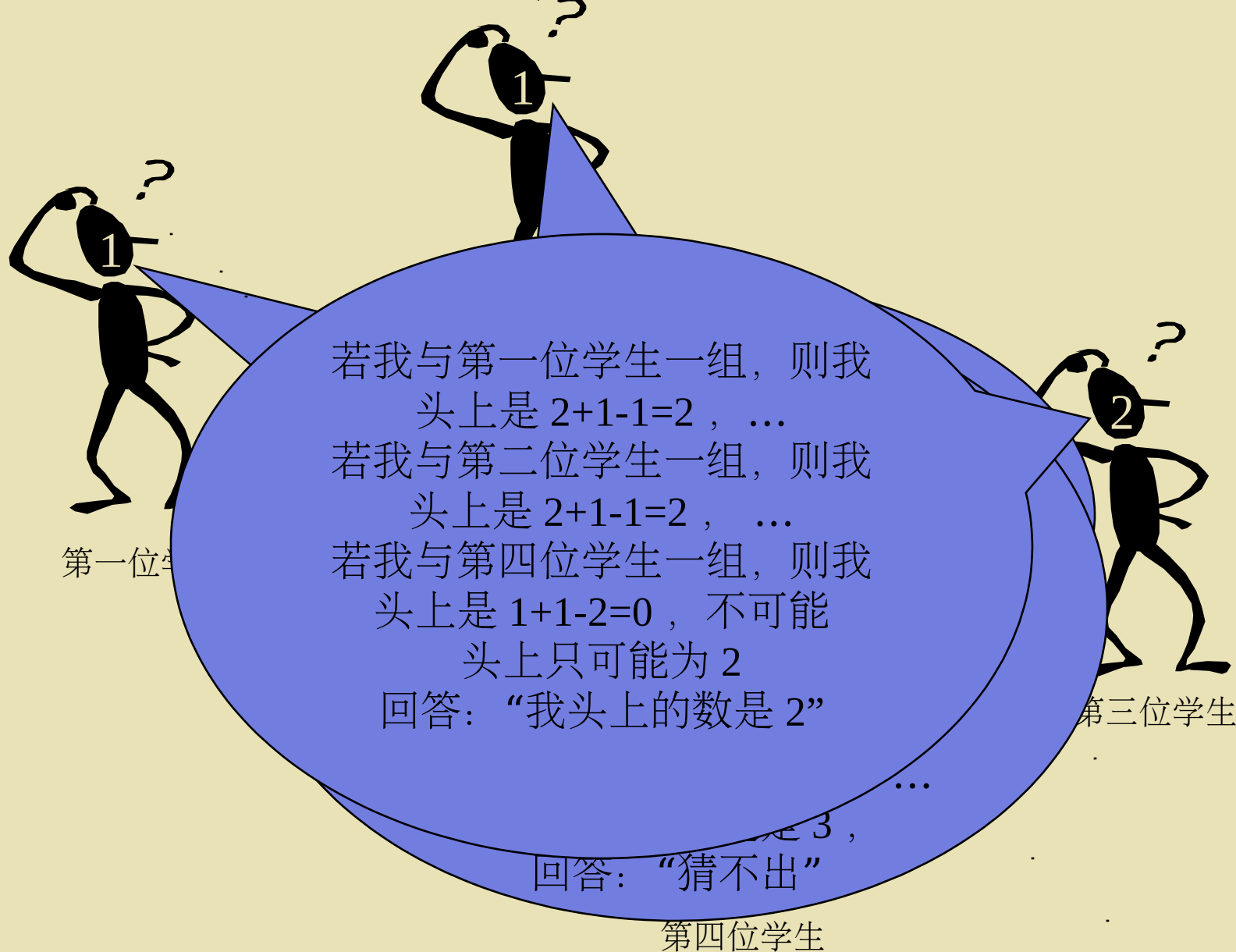
教授轮流向  $n$  位学生发问：是否能够猜出自己头上的数。经过若干次的提问之后，当教授再次询问某人时，此人突然露出了得意的笑容，把贴在自己头上的那个数准确无误的报了出来。

我们的问题就是：证明是否一定有人能够猜出自己头上的数，若有人能够猜出，则计算最早在第几次提问时有人先猜出头上的数，分析整个推理的过程，并总结出结论。

由于当  $n=3$  时， $m$  只能为 2，即为《聪明的学生》的问题原形，在论文中第一部分给出了证明。而对于  $m=n-1$  时的情况，作为原题的第一种推广情形，在论文中第二部分给出证明。下面着重讨论  $n>3$ ， $m<n-1$  时的情况。

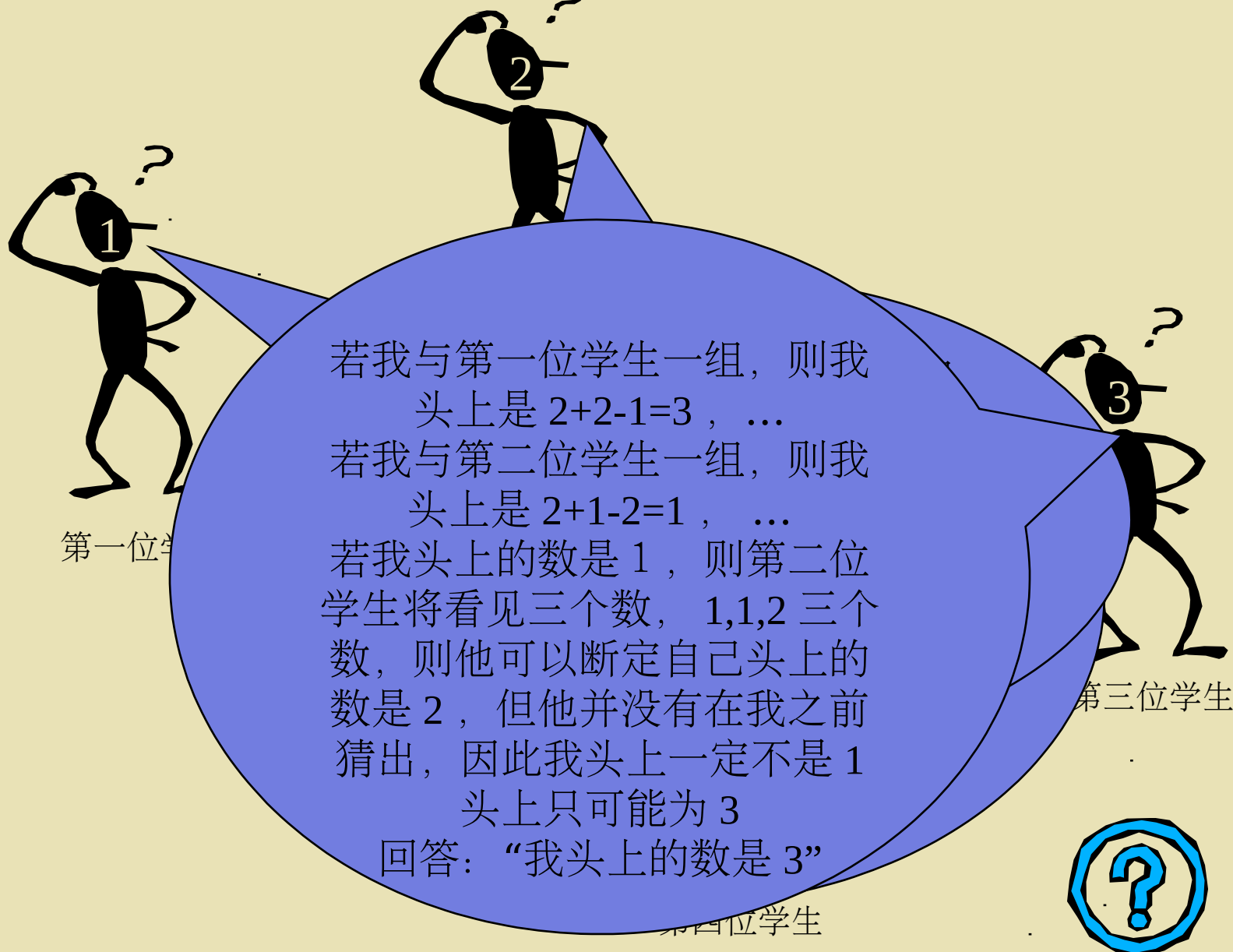



有 4 位学生，且每组有 2 人





有 4 位学生，且每组有 2 人





我们先来分析一下，如何能够猜出自己头上的数。当某位学生假设自己头上是某数，并通过推理发现别人理应能够在前  $n-1$  次提问中最先猜出头上的数，就可以根据实际上别人并没有在他之前猜出这一矛盾来排除头上是某数的情况。而当某位学生能够排除所有不同于实际的可能值时他便猜出自己头上的数了。

这样我们能够设计出最基本的算法，在每一次提问时，站在被提问的学生的角度去考虑问题，逐一考虑每个学生想法，直至某位学生能够猜出头上的数。



让我们来分析一下这样做的时间复杂度：

由于考虑可能的分组情况不同，头上的数就有可能不同。因此对于每位学生头上的数最多有  $C_n^m$  种不同情况。

因此对第  $N$  次提问，分析的复杂度为  $O((C_n^m)^N)$

可见，这样是不可能很好的解决问题的，我们必须从根本上去优化算法，唯一的途径就是从问题的本质去研究其中的联系，从而避开表面上繁琐的“思维嵌套”，即：在  $C$  的脑海中要考虑  $B$  是如何思考  $A$  的想法。





用  $n+1$  元组  $(A_1, A_2, \dots, A_n, k)$  来描述推理过程中的一种情形, 其中  $n$  个人头上数分别为  $A_1, A_2, \dots, A_n$ ; 从第一位学生开始提问, 且当前的被提问者为第  $k$  位学生。

在左边的例子中, 若对第一位学生进行提问, 对于  $n+1$  元组  $(1, 1, 2, 2, 1)$  来推理,  $k$  位学生可以不依靠别人的回答进行推理, 能够直接判断出自己头上数, 称  $n+1$  元组描述的情形为“终结情形”。

在左边的例子中, 若对第三位学生进行提问时, 第三位学生可以直接推理出头上的数是 2, 因此 5 元组  $(1, 1, 2, 2, 3)$  是“一类情形”。对于  $n+1$  元组  $(A_1, A_2, \dots, A_n, k)$ ,  $A_k$

$= \max\{A_1, A_2, \dots, A_n\}$ , 称  $n+1$  元组描述的情形为“二类情形”。

如  $(1, 1, 2, 2, 1)$ ,  $(1, 1, 2, 2, 2)$  为“一类情形”,  $(1, 1, 2, 2, 3)$ ,  $(1, 1, 2, 2, 4)$  为“二类情形”。

为第 k 位学生假设分组 X 情况下两组学生头上数的和。记 为所推测出的头上数的可能值。

下面例子中，对每位学生，都有三种不同的分组情况： $\{t_1; t_2\} \setminus \{k\}$

1. 一与二一组，三与四一组。 $t_1, t_2$

2. 一与三一组，二与四一组。 $t_m$

3. 一与四一组，二与三一组。

第一位学生在学组推測出第一组的头数为 G，另一位同学在第二组推測出 T，这两位同学头上的数之和为  $G+T=4$ ，由此可推出第一位学生的头数为 1，第二位学生的头数为 3。



在上例中，考虑第一位学生

$$M = \sum_{i=1}^{k-1} a_i + \sum_{i=k+1}^n a_i$$

$$M=1+2+2=5$$

$$G(T) = \sum_{i=1}^m At_i = \sum_{i=m+1}^{n-1} At_i + A'_k$$

$$G(T)=2+2=4$$

注意到  $m \geq n/2$ ，因此

$$A'_k = \sum_{i=1}^m At_i - \sum_{i=m+1}^{n-1} At_i \geq At_1$$

$$A'_k = 2+2-1=3 \geq 2$$

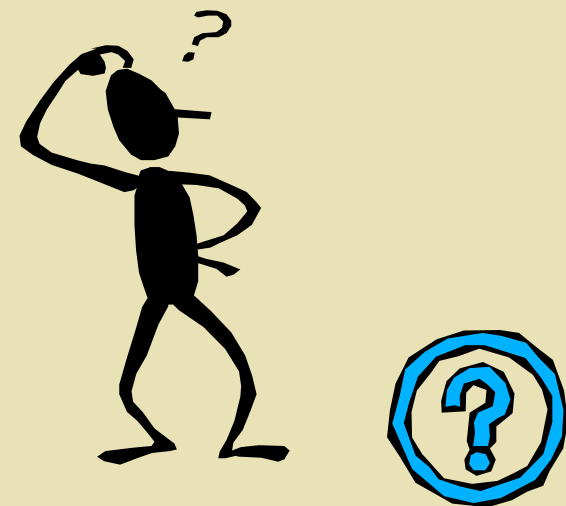
对于  $(A_1, A_2, \dots, A_n, k)$  任意

一个分组  $X$ ，设


$$\{x_1, x_2, \dots, x_{n-1}\} = \{1, 2, \dots, n\} \setminus \{k\}$$

显然有  $G(T) \geq G(X)$

$$A'_k = 2G(X) - M$$



每位学生由于不能直接判断头上一定是某数，就只能采取排除法。由于他们不会猜错，因此只需考虑他们如何能够排除头上数不同于实际的可能情况。显然只有通过推理导出矛盾，才能够得到有用的信息，排除某些可能情况。因此，问题的关键就在于分析如何导出矛盾。



我们已在定义中将能够不通过推理直接猜出头上的数的情况，称之为“终结情形”。考虑上面的分析，若某人处于“终结情形”，而他却没有猜出头上的数，这显然是一个矛盾。而对于不是“终结情形”的情况，则需要进行推理。因此问题转变为考虑怎样的情况为“终结情形”，以及怎样的情况能够通过推理转变为“终结情形”，可见解决问题的关键，在于深入分析“终结情形”。



而我们之所以在定义中将所有学生分成了两类考虑：“一类情形”及“二类情形”，是受到原问题《聪明的学生》的启发，虽然在推广的问题中，头上数最大的学生不再处于十分特殊的地位，即头上数为其余学生头上数的和，并且可能不止一人头上的数为最大数。但我们有理由相信这依然是解决问题的一个入手点。



考虑“终结情形”的条件:

➤当  $(A_1, A_2, \dots, A_n, k)$  为“一类情形”

考虑在分组  $T$  的情况下

$$A_k < \max\{A_1, A_2, \dots, A_n\} = At_1 \leq A'_k$$

这种情况不同于实际情况, 需要进行推理。得到结论: “一类情形”不可能为“终结情形”。

➤当  $(A_1, A_2, \dots, A_n, k)$  为“二类情形”

要使得  $(A_1, A_2, \dots, A_n, k)$  为终结情形, 必然要满足在分组

$T$  的情况下  $A_k = A_k$ 。因此实际分组  $B$  满足  $\sum_{i=1}^m Ab_i = \sum_{i=1}^m At_i$ 。  
 , 因此可得  $G(B) = G(T)$

显然  $G(X) \leq G(T) = G(B)$ , 在此条件下, 若存在可能的分组

$C$  满足  $G(C) < G(B)$   $A'_k = 2G(C) - M < 2G(B) - M = A_k$

$(A_1, A_2, \dots, A_n, k)$  显然由于  $A'_k \leq 0$   $2G(C) - M \leq 0$

若不存在可能的分组  $C$  满足  $G(C) < G(B)$ , 必然有即任意分组  $G(X) = G(B)$

即  $At_1 = At_2 = \dots = At_{n-1}$

进一步分析对可能的分组  $C$  满足  $G(C) < G(B)$   
 , 何时满足  $2G(C) - M \leq 0$  , 分两种情况考虑

:

➤ 当  $At_2, At_3, \dots, At_{n-1}$  不全相等时, 必然有  $At_2 > At_{n-1}$   
 , 考虑如下分组  $C$ : 第  $t_1, t_3, t_4, \dots, t_{m-1}, t_m, t_{n-1}$  位学

生为  $t_2, t_{m+1}, t_{m+2}, \dots, t_{n-2}$  组, 第  $t_1, t_3, t_4, \dots, t_{m-1}, t_m, t_{n-1}$  位学  
 $G(C) < G(B)$

位学生为一组, 此时显然有

需满足  $2G(C) - M \leq 0$  , 因此  $At_1 + \sum_{i=3}^m At_i + At_{n-1} \leq At_2 + \sum_{i=m+1}^{n-2} At_i$

而  $At_1 + \sum_{i=3}^m At_i + At_{n-1} > At_1 + \sum_{i=3}^m At_i \geq At_1 + \sum_{i=3}^{n-m} At_i \geq At_2 + \sum_{i=m+1}^{n-2} At_i$

矛盾。因此, 若  $At_2 > At_{n-1}$   $(A_1, A_2, \dots, A_n, k)$  不可能为终  
 结情形。



➤ 当  $At_2, At_3, \dots, At_{n-1}$  全相等时, 不妨设  $At_1 = p$ ,

$$At_i = q, i \in \{2, 3, \dots, n-1\}$$

考虑分组 C:  $m$  个头上数为  $q$  的学生为一组, 剩下的学生为一组 (第  $k$  位学生在这组中)。

由于  $2G(C) - M \leq 0$ , 因此  $(2m - n + 2)q \leq p$ 。

❖ 当  $m > n/2$  时, 再来考虑分组 D: 头上数是  $p$  的学生与  $n-m-1$  个头上数是  $q$  的学生在一组, 剩下学生为一组。  
 $G(D) < G(B)$

由于  $2G(D) - M \leq 0$ , 因此  $p \leq (2m - n)q$ , 观察得到的两个不等式, 显然矛盾, 因此当  $m > n/2$  时, 若  $At_1 > At_{n-1}$  不可能为终结情形。

❖ 当  $m = n/2$  时, 可以得到  $A_k = \sum_{i=1}^m At_i - \sum_{i=m+1}^{n-1} At_i = At_1 = p$  且  $2q \leq p$  且。

因此终结情形的条件为

➤  $At_1 = At_2 = \cdots = At_{n-1}$

➤ 或  $(A_1, A_2, \cdots, A_n, k)$  为二类情形, 且

$$m = n/2, p, q \in \mathbb{Z}^+, 2q \leq p, A_k = At_1 = p, At_i = q, i \in \{2, 3, \cdots, n-1\}$$

有 4 位学生  
，且每组有  
2 人



在左例中，可以用上述结论，推出第三、四位学生可以不依靠别人的回答进行推理，直接猜出头上的数



我们可以考虑所有能够猜出自己头上数的“一类情形”中所用猜测次数最少的情况，利用极端性原则，用反证法可以证明，任何“一类情形”的被提问者都不可能最先猜出自己头上的数。由于证明过程非常繁琐，考虑到时间关系，这里将不给出详细的证明过程。

由于推理过程中是利用产生矛盾来得出结果的，因此我们利用上面这一结论，可以忽略推理过程中所有的“一类情形”，即我们只需考虑头上数最大的学生的想法。





由于关于这一部分的证明极其繁琐，考虑到时间关系，下面只能给出一些结论

- 若  $n$  个学生头上的数都相等，则第一位学生可以猜出头上的数。
- 若  $n$  个学生头上的数不全相等
  - ❖ 若有三个以上的最大数或  $m > n/2$ ，则没有人能够猜出头上的数
  - ❖ 若有两个最大数
    - 若  $n=4$ ，则必然是头上数最大的人猜出头上的数，但需要通过推理才能确定由哪一人
    - 若  $n>4$ ，则没有人能够猜出头上的数
  - ❖ 若只有一个最大数，则需进行推理

若推理过程中出现“终结情形”，则可以直接得到结果，否则需要考虑每一种可能的分组情况，对有可能猜出头上数的学生，必然要满足如下条件：


➤  $A_k = 2G(T) - M$

- 任意可能分组  $C$  符合  $G(C) < G(B)$ ，满足  $At_1$  且推理过程中， $n$  个学生头上的数始终在减小



虽然我们对问题进行了深入的分析，对推理加强了判定，对算法本质进行了优化，将解决问题的时间复杂度大幅度下降，但依然可能出现多种考虑情况。所以推理已不像《聪明的学生》那样是一个线性结构的推理，整个推理过程将成为树状结构。因此我们除了在算法本质上进行优化的同时，加强在编程实现的优化，由于考虑到我们总结出的推理方式中， $n$  个学生头上的数始终在减小，因此我们可以采用纪录搜索的方式解决问题，为了加快检索速度，可以采用 Hash 表，并且我们为了方便描述一个情形，即  $n$  位学生头上的数，可以采用一个  $n$  位  $p$  进制数来描述。综合两方面的优化，我们已经较为圆满的解决了这个问题。






我们深入地分析了一个逻辑推理问题，从综观全局的角度来考虑问题的本质联系，而非一味单纯地从每个人的思想出发，简化了最烦琐的“思维嵌套”，因此避免了问题规模随着推理次数急剧增长，有效地解决了问题。

从基本的例子入手分析→考虑问题的本质→从本质入手分析矛盾→考虑何种情形为“终结情形”→考虑何种情形能归结到“终结情形”→分情况讨论并加以证明→得出结论并形成算法

相信对这样一道问题的解决，对处理烦琐的“思维嵌套”问题提供了一种可以借鉴的方法。





# 谢谢！

请多多指教