一类猜数问题的研究

长沙市雅礼中学 龙凡

猜数问题

- 猜数问题是信息学竞赛中一种常见的类似博弈的问题。
- 基本形式：存在一个被猜数 X。每次可以猜一个数 Y，之后会返回 X 和 Y 的关系，你要利用返回的信息来猜出 X。
- 本文着重讨论的一类猜数问题是：返回的信息是 X 和 Y 的大小关系。

基本的猜数问题

- 下面就是一个本文讨论范畴内的最基本的猜数问题：
- 被猜数 X 是 1 到 N 范围内的整数，每次你可以询问一个整数 Y 和 X 的大小关系。给出 N ，请问在最坏情况下至少需要几次才能保证猜出 X 。
- 上题应用大家熟悉的二分的思想可以轻松解决。

基本的猜数问题

- 设 N=7,X=5 。应用二分的思想询问:

1	2	3	4	5	6	7

1. 询问 Y=4
2. 得到 X>Y
3. 询问 Y=6
4. 得到 X<Y
5. 询问 Y=5
6. 得到 X=Y

上[illegible]次询问。这只是作为[illegible]。对于此题，应用二分法，作适当的数学分析，就可以得到最少的询问次数为：

$$\lceil \log_2(n+1) \rceil$$

X=5!!

问题的提出

- 作为一类猜数问题中最基本的形式，固然是非常简单的。
- 但很多其他的问题在这个“基本形式”上进行了一些扩充和加强，就变成了非常棘手的问题。

- 下面就让我们来实际的看一看这类型的例子：

问题的提出

　你和你的同学在网上进行猜数游戏，基本规则和普通的猜数游戏一样。你猜一个数Y，他告诉你，Y和他心中所想的数X的大小关系。你打字速度很快，每1s钟就可以问一个问题。由于网络有延迟的，所以你问的问题不能马上得到回答，而是要经过1s之后才会得到回答。同时你和同学约定，你要尽量避免得到X>Y的回答。（你可以认为，X是他的考试成绩之类的☺）当你累计得到第K次X>Y的回答的时候，你就输了。已知X是1到N之间的整数。现在给出N、K，请问最少要多少秒才能保证猜出X。

问题的提出

- 下面是一个 N=6,K=3 时的实例：

询问	回答	时间
Y=3	…	1s
Y=5	X>3	2s
Y=6	X<5	3s
Y=4	X<6	4s
Y=4	X=4	5s

- 上面的询问，总共用了 5s 。
- 期间得到了一次 X>Y 的回答。K=3 没有超过限制。

初步分析

- 我们现在的问题并不是如何猜，而是在最坏的情况下至少要多少秒才能猜出 X 来。
- 本题沿用二分的老思路是行不通的。
- 可以看出，虽然仅仅是在基本猜数问题上进行了些许加强，就成了一个非常棘手的问题。

当 N=3，K=1 并且 X=3 时。

1	2	3

- 为了解决这个问题，让我们重新从最原始的猜数问题开始分析。

询问 Y=2，必然得到 X>Y 回答。

再看基本猜数问题

- 前面我们是通过二分的方法来解决此题的。至于“二分”这个思路的来源，更多的是源自猜测、及平时做题的经验。
- 下面就来系统的分析为什么“二分”是正确的。

被猜数 X 是 1 到 N 范围内的整数，每次你可以询问一个整数 Y 和 X 的大小关系，系统会给出答案，请问在最坏情况下至少需要询问多少次才能保证猜出 X。

- 通过分析，希望能找到一个更具有普遍性的方法来解决前面的题目。
- 让我们尝试用递推的方法来分析问题。

再看基本猜数问题

- 设 f(i) 表示 i 次询问最大能够处理的区间长度。即若 N=f(i) ，则只需要 i 次就可以猜出 X 。
- 根据上面定义，若 f(j)≥N 。且 f(j-1)<N 则可以知道 j 即为题目所求的至少询问次数。
- 每询问一个 Y ，只可能得到三种不同的回答，我们就从这三种不同的回答入手来分析问题。

再看基本猜数问题

- 询问一个值 Y。
- 若回答是 X=Y ，即游戏马上中止。
- X>Y 时，为了要在剩下的 I-1 次询问中得到答案，大于 Y 的区域长度不能超过 f(i-1)。
- X<Y 时，小于 Y 的区域长度不能超过 f(i-1)。

小于 Y 的区域	Y	大于 Y 的区域
f(i-1)	1	f(i-1)

$$f(i)=2f(i-1)+1$$

再看基本猜数问题

- 上面分析直观的说明了二分的正确性。
- $f(i)=2f(i-1)+1 \Rightarrow f(i)=2^i-1$
- 应用递推的方法间接的证明了前面的结论，即最少询问的次数为：

$$\lceil \log_2(n+1) \rceil$$

- 更重要的是：对于这类试题来说，上面这种应用递推的分析问题的方法具有很强的推广性。

二次分析

- 通过上面的分析，基本的猜数问题已经完整的解决了。现在回到原题的研究。
- 现在直接分析原题仍然有点困难，注意到原题相比基本猜数问题有两个加强：

  1. 累计获得 K 次 X>Y 的答案就游戏结束。

  2. 本次的回答要在下一个提问之后获得。
- 不妨先把这道题目拆开，分部解决。

猜数问题的加强

- 被猜数 X 是 1 到 N 范围内的整数，每次你可以询问一个整数 Y 和 X 的大小关系。
- 附加上你要尽量避免询问 X<Y 的值，因为累计获得 K 次 X>Y 的答案就游戏结束。
- 给出 N 和 K，问在最坏情况下至少需要询问几次才能保证猜出 X。
- 对于这个问题，可以借鉴前面的经验，应用递推的方法来求解。

猜数问题的加强

- 类似的，我们设 f(i,j) 表示用 i 次询问，在累计 j 次 X>Y 的回答就游戏结束的情况下，最大能处理的区间长度。
- 如果我们能够快速求出 f ，问题也就容易解决了。找到最小的数 Ans ，满足 $f(Ans,k)\geq N$ ， $f(Ans-1,K)<N$ 。则答案就为 Ans 。
- 和上一题类似，可以根据三种不同的回答，即： X=Y ， X<Y ， X>Y 分别进行讨论来构造递推公式。

猜数问题的加强

小于 Y 的区域	Y	大于 Y 的区域
f(i-1,j)	1	f(i-1,j-1)

- X=Y 的情况最简单，长度为 1 。
- 回答为 X<Y ，还剩下 i-1 次机会询问，所以小于 Y 的区域长度最多为 f(i-1,j) 。
- 回答为 X>Y ，还剩下 i-1 次机会询问，并且我们得到了一次“ X>Y” 的回答，所以 j 值减少 1 ，大于 Y 的区域长度最多为 f(i-1,j-1) 。

猜数问题的加强

- 根据这个递推式，依次将 f 全部计算出来，直到存在一个 f(Ans,k)≥N ，就结束。所以总时间复杂度为 O(AnsK) 。
- 这样此题就被我们解决了。但是如果我们不是要求询问次数，而是希望得到每一步询问的策略，该怎么办呢？
- 为了让大家深入理解递推算法，我们在这里做进一步分析。

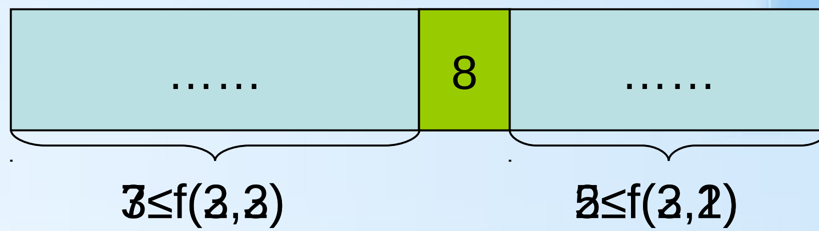
猜数问题的加强

小于 Y 的区域	Y	大于 Y 的区域
f(i-1,j)	1	f(i-1,j-1)

- 仔细看上图。事实上询问的值 Y 已经提示出来了，Y 是第 f(i-1,j)+1 个可能值。
- 在还剩下 i 次询问，并累计 j 次 X>Y 回答就游戏结束的情况下，若 X 可能的出现区间是 [L,R] ，则 Y=L+f(i-1,j) 。
- 下面是一个简单的例子：

例子

f(i,j)	j=1	j=2	j=3
i=1	1	1	1
i=2	2	3	3
i=3	3	6	7
i=4	4	10	14

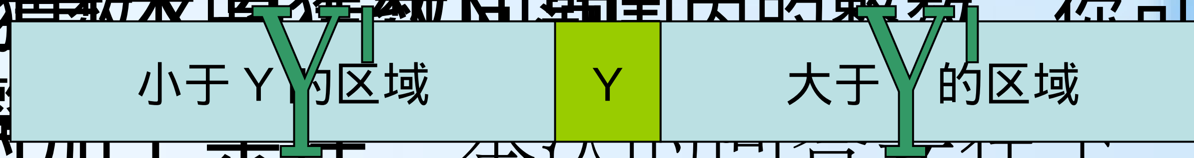


无论得到什么回答，都能解决！

- 上表是已经计算好的 f 值。
- 若 N=6 ， K=2 ，查上表可以得出需要 3 次询问才能保证猜出 X 。第一次询问 Y=4 。
- 若 N=13,K=3 ，查上表可以得出需要 4 次询问才能保证猜出 X 。第一次询问 Y=8 。

猜数问题的加强 2

- 被猜数 ……围内的整数，你可以询问一个数……问题不同，提问的回答要在下一次提问之后才能获得。给出 N，问最坏情况下需要多少次询问才能猜出 X。



- 附加了条件：……问之后获得。

- 本题最大的难点是：当询问了一个值 Y 之后，下一次询问之前我们得不到任何的讯息，必须盲目的再询问一个值 Y’。
- 其实本题仍然可以用递推的思想解决。

猜数问题的加强 2

- 我们设 f(i) 表示 i 次询问能够处理的区间最大的长度。
- 则可以知道若 f(ans)≥N ， f(ans-1)<N 则可知 Ans 即为所求的最少的询问次数。

- 由于没有其他条件的限制，大于 Y 和小于 Y 两个区域是等价的。不妨设我们将 Y’ 询问在了小于 Y 的区域：

猜数问题的加强 2

- 我们先询问 Y ，然后再询问 Y’。这时才能得到关于 Y 的回答。
- 若 X>Y ，则关于 Y’ 的询问被浪费了。还剩下 i-2 次询问机会。大于 Y 的区域的最大为 f(i-2) 。
- 若 X<Y ，则 Y’ 的询问没有被浪费，还剩下 i-1 次询问机会。小于 Y 的区域最大为 f(i-1) 。

小于 Y 的区域	Y’	小于 Y 的区域	Y	大于 Y 的区域

$$f(i) = f(i-1) + f(i-2) + 1$$

$$\underbrace{\text{小于 Y 的区域} \quad Y' \quad \text{小于 Y 的区域}}_{f(i-1)} \quad Y \quad \underbrace{\text{大于 Y 的区域}}_{f(i-2)}$$

原问题的解决

- 上题的时间复杂度是 O(Ans) 。
- 至此两个加强问题，已经被分别解决。
- 有了解决这两个基础问题的过程，再回来看原问题就比较容易了。

被猜数 X 是 1 到 N 范围内的整数，你可以询问一个整数 Y 和 X 的大小关系。与普通猜数问题不同，提问的回答要在下一次提问之后才能获得。你要尽量避免询问比 X 小的 Y 值，因为累计获得 K 次 X>Y 的答案就游戏结束。给出 N 、 K ，问在最坏情况下至少需要询问几次才能保证猜出 X 。

原问题的解决

- 设 $f(i,j)$ 表示用 i 次询问，在用计 j 次获得为 Y 的回答就游戏结束的情况下，最大能处理的区间长度。

小于 Y 的区域	Y’	小于 Y 的区域	Y	大于 Y 的区域

$f(i-1,j)$　　$f(i-2,j-2)$

- 先询问 Y ，再 Y’。将 Y’ 询问在小于 Y 的区域。
- 若 X>Y ，则 Y’ 的询问浪费了，询问次数减二，且会获得两次 X>Y 的回答，j 值减二，所以大于 Y 的区域最长为 f(i-2,j-2) 。
- 若 X<Y ，询问次数减一。不会得到 X>Y 的回答，所以 j 不变。小于 Y 的区域最长为 f(i-1,j) 。

原问题的解决

- 做上面的分析之后是不是就足够了呢？
- [illegible]的区域和大于 Y 的区域不再等价，我们必须讨论 Y' 询问在大于 Y 的区域时的情况。

小于 Y 的区域	Y	大于 Y 的区域	Y'	大于 Y 的区域

$f(i-2,j)$ $f(i-1,j-1)$

- 将 Y' 询问在大于 Y 的区域。
- 若 X<Y ，Y' 的询问被浪费了，询问次数减少二。且不会获得 X>Y 的回答，j 值不变。小于 Y 的区域最长长度为 f(i-2,j) 。
- 若 X>Y ，询问次数减少 1 。得到了 X>Y 的回答，j 减少 1 。大于 Y 的区域最长为 f(i-1,j-1) 。

原问题的解决

- 综合上面，可以得出动态规划公式：

$$f(i,j)=\max\begin{cases} f(i-1,j)+f(i-2,j-2)+1 \\ f(i-1,j-1)+f(i-2,j)+1 \end{cases}$$

- 边界条件为：

$$\begin{cases} f(i,0)=0 \\ f(0,j)=0 \\ f(1,j)=0 \end{cases}$$

原问题的解决

- 算法的框架就是利用得出的动态规划转移公式，依次计算 f 值。直到有一个 Ans 满足 f(ans,k)≥N,f(ans-1,k)<N 。则所求的最少次数为 Ans 。
- 这个算法的总时间复杂度为 O(AnsK) ，由于 f 函数的增长非常快，所以实际运行效果很理想。
- 至此我们完整的解决了文初的问题。

总结

- 同类的试题还有很多，这里只是列举了其中的典型。
- 在本文里主要介绍的是如何使用递推、动态规划的方法来解决同类猜数问题。这些算法应用在猜数问题上，能够更加深入的、系统的分析问题。
- 不仅仅局限于同类问题，只要我们学会如何抛开繁杂的题目描述，从数学模型的角度系统的来分析问题 。任何难题都能迎刃而解。