

apex解题报告

福州一中 卓亮

April 30, 2012

Contents

1	题意简述	2
2	命题思路	2
3	考查点	2
4	问题分析	2
5	数据生成方法	4
6	选手得分统计	4
7	后记	5
	Appendices	7
A	apex完整题面	7

1 题意简述

定义数列 $x_1 = 1, x_{2k} = -x_k, x_{2k-1} = (-1)^{k+1}x_k (k \geq 1)$, 设 $S_n = \sum_{i=1}^n x_i$ 。有以下3种询问。

1. 求 x_n ;
2. 判断 S_n 的符号;
3. 求 S_n 。

2 命题思路

本题是本人在做一道数学题时想出的。一开始打算利用程序来找规律，后来灵机一动，弄出了一个重要的等式。虽然那道数学题没有完整解出来，不过本人想到可以利用这个等式编一道有意思的题目。

这3问设计得很有层次性，即使某一问不会做，也很有可能做出其他的询问。而这3问又具有联系。

3 考查点

本题主要考察了选手代数方面的能力，以及分析问题，寻找规律的能力。

4 问题分析

这类问题通常有两种思路。一种是预处理出所有答案，然后碰到询问直接回答；另一种是对每个询问单独处理。

如果采用第一种思路，我们可以考虑预处理出 x_1, x_2, \dots, x_n ，以及 S_1, S_2, \dots, S_n ，然后就可以以 $O(1)$ 的时间复杂度回答每个询问。预处理出 x_1, x_2, \dots, x_n ，可以直接利用数列定义式来完成，时间复杂度是 $O(n)$ 。而预处理出前 n 项和 S_n 的时间复杂度也是 $O(n)$ 的。这个算法的期望得分是50分。

由于预处理出 n 以内的所有答案的时间复杂度下限就是 $\Omega(n)$ ，这种思路在 n 很大时无法胜任。因而我们考虑第二种思路。

第一问十分简单。我们直接利用题目所给的数列定义式，就可以求出 x_n 的值了。这一问的时间复杂度是 $O(\log n)$ 。

再看第二问，容易发现，如果解决了第三问，那么第二问也解决了。因而先尝试解决第三问。

考虑 $S_{4n}(n \geq 1)$ 。根据 $x_{2k} = -x_k, x_{2k-1} = (-1)^{k+1}x_k(k \geq 1)$,

$$\begin{aligned}
 S_{4n} &= \sum_{i=1}^{4n} x_i \\
 &= \sum_{i=1}^{2n} x_{2i-1} + \sum_{i=1}^{2n} x_{2i} \\
 &= \sum_{i=1}^{2n} (-1)^{i+1} x_i - \sum_{i=1}^{2n} x_i \\
 &= \sum_{i=1}^{2n} ((-1)^{i+1} x_i - x_i) \\
 &= \sum_{i=1}^n -2x_{2i} \\
 &= 2 \sum_{i=1}^n x_i \\
 &= 2S_n
 \end{aligned} \tag{1}$$

由式(1)，立即得到一个快速求 S_n 的做法：若 $4 \nmid n$ ，那么 $S_n = S_{n-1} + x_n$ ；否则， $S_n = 2S_{n/4}$ 。这个算法的时间复杂度由下式给出： $T(n) = T(n/4) + O(\log n)$ ，解得 $T(n) = O(\log^2 n)$ 。

注意到这个算法有系数3，即每次除以4，最多要3轮。我们还可以对此进行一些优化。注意到

$$x_{4n+1} + x_{4n+2} = (-1)^{2n+2}x_{2n+1} - x_{2n+1} = 0 \tag{2}$$

我们可以得到

$$S_{4n+2} = S_{4n} + x_{4n+1} + x_{4n+2} = S_{4n} \tag{3}$$

这样，我们在每一次除以4的时候最多只要算1次 x_n ，就消除了系数3。

有了第三问，第二问显然也可以做到 $O(\log^2 n)$ 。不过我们有更优的算法。首先，可以用数学归纳法证明， $S_n \geq 0$ 。因而，我们只需要找出哪些位置是0。

容易看出, $S_{2n-1} \neq 0$ 。这是因为, 数列 $\{x_n\}$ 只含数字1和-1。由于 $S_{i+1} = S_i + x_{i+1}$, 因此 S_{i+1} 与 S_i 奇偶性不同。而 $S_1 = 1$ 为奇数, S_{2n-1} 与 S_1 奇偶性相同, 因而也为奇数。而0是偶数, 故而 $S_{2n-1} \neq 0$ 。

根据式(1)(3), 以及 $S_2 = 0, S_4 \neq 0$, 可以看出 $S_n = 0$ 当且仅当 n 的4进制表示仅含0和2。这样第二问就做到了 $O(\log n)$ 。

5 数据生成方法

本题的数据生成方法十分容易, 只需要依据数据范围, 直接随机生成即可。不过对于询问2, 仍然遇到了一点问题。

注意到我们在上述分析中提到了询问2, 回答是0的条件。在1至 10^{18} 内, 回答是0的数的个数大概只有 2^{29} 。当我们在1至 10^{18} 范围内随机时, 随机到0的概率小于 10^{-9} 。即使随机 10^5 次, 存在一个回答是0的数的概率仍小于0.01%。事实上, 对于大的随机数据, 所有询问2的回答都是正数。

因为发生了这种情况, 本人对大数据询问2的生成进行了一些改动, 手造了一些回答是0的情况, 混合进去, 从而避免了全部是正数导致的数据的不全面。

6 选手得分统计

本题对国家队候选队员进行了测试, 得分统计如表(1)所示。

分数	人数
20	1
60	1
65	2
100	7

Table 1: 得分统计

本题的平均分大约为82.7分。

7 后记

本题是一个数列问题。正如来自南京师大附中的顾昱洲同学所言，数列问题往往可以用oeis来解决。这里，oeis，可参考[1]，指的是一个整数数列网站，收录了数以万计的数列，包含了丰富的数列的性质。如果本题可以轻易地用搜索工具搜索到性质，甚至通项，那么本题就显得十分没有意义了。有鉴于此，本人对此数列进行搜索，幸运的是，这个网站根本没有这个数列。利用知名的搜索引擎进行搜索，也没有发现任何有关这个数列的信息。因而，将此题命制出来十分妥当，没有什么可以钻空子的地方。

本题进行测试后，收到了很好的效果。绝大多数选手都拿到了十分理想的分数，而选手们采取的方法也是花样繁复，争奇斗艳。

来自杭州外国语学校的陈立杰同学，将本文中提到的第三问解法优化至 $O(\log n)$ 。事实上，只要对中途 x_n 的询问实施记忆化即可。下面，我们从另一个角度说明这样做的时间复杂度。

我们每一步，同时计算2个量： S_n, x_{n+1} 。由 x_{n+1} 可以推出 x_{2n+1} 和 x_{2n+2} ，由此又可以推出 $x_{4n+1}, x_{4n+2}, x_{4n+3}, x_{4n+4}$ 。利用以上四个量，就可以轻易地由 S_n 推出 $S_{4n}, S_{4n+1}, S_{4n+2}, S_{4n+3}$ 了。故，求 S_n, x_{n+1} ，可以化归为求 $S_{\lfloor n/4 \rfloor}, x_{\lfloor n/4 \rfloor + 1}$ 。这样 $T(n) = T(n/4) + O(1)$ ，因此 $T(n) = O(\log n)$ 。

References

- [1] *The On-Line Encyclopedia of Integer SequencesTM (OEISTM)*[EB/OL]. <http://oeis.org/>.

Appendices

A apex完整题面

Time Limit: 1s, Memory Limit: 128MB

定义数列 $x_1 = 1, x_{2k} = -x_k, x_{2k-1} = (-1)^{k+1}x_k (k \geq 1)$. 存在如下三种类型询问.

1. 求 x_n .
2. 判断 $x_1 + x_2 + \dots + x_n$ 的符号.
3. 求 $x_1 + x_2 + \dots + x_n$.

Task: apex

Input: stdin

第一行包含一个整数 q , 代表询问的数量. 接下来 q 行, 每行两个数 c, n , 即询问的类型, 以及 n 的值。

Output: stdout

对于每个询问, 输出一行, 即对应的答案. 对于询问2, 如果是正数, 输出“+”, 如果是负数, 输出“-”, 如果是0, 输出“0”。

Sample Input	Sample Output
3	1
1 1	0
2 2	1
3 3	

Constraints

- 有10%的数据, 仅含询问1.
- 有10%的数据, 仅含询问2.
- 有10%的数据, 仅含询问3.
- 有10%的数据, 仅含询问1和2.

有10%的数据, 仅含询问2和3.

有10%的数据, 仅含询问1和3.

对于50%的数据, $1 \leq n \leq 10^7$.

对于100%的数据, $1 \leq n \leq 10^{18}, 1 \leq q \leq 10^5$.