

# Bazarek 解题报告

长郡中学 陈胤伯

## 1 试题来源

<http://main.edu.pl/en/archive/pa/2014/baz>

## 2 试题大意

有一棵  $n$  个点的有根树，每个非叶子节点有奇数个儿子。

每个叶子节点有三种状态，要么被染成了红色，要么被染成了蓝色，要么未被染色。

小红和小蓝在上面玩游戏，他们轮流操作（小红先手），每次操作者可以选择一个未被染色的叶子染上自己的颜色。

当所有叶子都有颜色之后，整棵树的所有非叶子节点会按逆BFS序确定颜色，即每个点变成它儿子中出现次数更多的那一种颜色。

最终根节点的颜色代表胜利的一方。求先手是否必胜。如果先手必胜，还需要输出先手第一步可以选择叶子使得自己必胜。

$n \leq 1000$ 。

## 3 算法介绍

### 3.1 求出先手是否必胜

我们先不考虑求方案，只关心哪一方必胜。

由于树是一个递归的结构，我们不妨从基本元素考虑起。

考虑一个叶子，有三种状态：未染色、红色、蓝色。我们如果把它看做一个“局面”，那么原来三种状态对应的是：先手必胜、红必胜、蓝必胜。

那么是不是任意子树都只有这三种情况呢？

考虑子树  $u$ ，假如  $u$  的儿子都能表示成这三种情况，我们把每个儿子看成独立的游戏。

对于红必胜的子树，红自然不需要主动进去玩。蓝也不会进去玩，因为这个子游戏不是先手必胜而是红必胜，红只要做到“蓝玩我就玩”即可。由此可知红必胜最终还是红必胜。

那么蓝必胜自然也就同理了。

再考虑先手必胜的子树，谁先玩一下就会变成谁必胜的局面，之后再玩这棵子树就是“垃圾时间”了。

由此我们得出算法，不妨设先手必胜、红必胜、蓝必胜的子树个数分别为  $f$ 、 $a$ 、 $b$ ，由于小红小蓝是轮流玩，每次玩的人都会占领一棵先手必胜的子树，那么这棵子树：

- 如果  $a - b > f \bmod 2$ ，则红必胜；
- 如果  $b - a > f \bmod 2$ ，则蓝必胜；
- 否则先手必胜。

这样每棵子树最终都能简化为这三种情况中的一种，我们用一个  $O(n)$  的自底向上的递推就能方便地求出。

### 3.2 求出方案

注意到原题目中  $n \leq 1000$ ，我们直接枚举  $O(n)$  先手第一步染哪个叶子，再  $O(n)$  检验此时根节点是否为红必胜或先手必胜。

复杂度  $O(n^2)$ 。

### 3.3 更好的做法

明明  $O(n)$  求出了每个子树的状态，却要  $O(n^2)$  求出所有可行方案，我们不禁觉得上述做法实在是 naive。

由于瓶颈在于求方案的  $O(n^2)$ ，我们考虑从最初递推的思路出发，优化这个过程。

### 3.3.1 $\text{redWin}(u)$

定义过程  $\text{redWin}(u)$  表示：子树  $u$ ，我希望先手（红）必胜，求出先手第一步可以选哪些叶子。

如果子树  $u$  是红必胜，那么  $\text{redWin}(u) = \text{all}(u)$ ， $\text{all}(u)$  表示  $u$  的全体叶子。

如果子树  $u$  是蓝必胜，那么  $\text{redWin}(u) = \emptyset$ 。

如果子树  $u$  是先手必胜，那么，若  $u$  是叶子则  $\text{redWin}(u) = \{u\}$ ，否则，我们考虑  $u$  的每个儿子  $v$ 。

回顾之前递推的思路，如果子树  $v$  是先手必胜，那么第一步占领  $v$  肯定是红色胜利的一种方案，于是加入  $\text{redWin}(v)$ 。

如果子树  $v$  是红必胜，小红第一步一定不会进去玩，因为这样等价于把先手让给小蓝。

如果子树  $v$  是蓝必胜，小红要想第一步进去玩还能确保胜利，那么一定要进去玩出一些优势，也就是把子树  $v$  从“蓝必胜”玩成“先手必胜”，这时小蓝要么得把子树  $v$  玩回原样，要么去抢另一个  $u$  的先手必胜的子树，不论是哪一种，子树  $u$  的局面回到小红先手时依然还是先手必胜。所以我们需要加入  $\text{firstWin}(v)$ 。

### 3.3.2 $\text{firstWin}(u)$

上面说了  $\text{firstWin}(u)$  表示：蓝必胜的子树  $u$ ，求先手（红）第一步可以选哪些叶子使得  $u$  变成先手必胜。

若  $u$  是叶子，那么  $\text{firstWin}(u) = \emptyset$ 。

设子树  $u$  先手必胜、红必胜、蓝必胜的子树个数分别为  $f$ 、 $a$ 、 $b$ 。若  $f \bmod 2 = 1$  是没有希望翻盘的，因为既然最终是蓝必胜，那么必有  $b \geq a + 1$ ，而根据奇偶性我们知道  $b \neq a + 1$ ，因此  $b > a + 1$ ，那么小红即使占领了  $u$  的一棵子树，结局依然是蓝必胜。若  $f \bmod 2 = 0$ ，因为是蓝必胜所以  $b > a$ ，因为小红得能够推翻“蓝必胜”所以  $b < a + 2$ ，因此  $b = a + 1$  时才有翻盘可能。

这样若  $f \bmod 2 = 0$  且  $b = a + 1$ ，我们枚举  $u$  的每个子树  $v$ ，若  $v$  是先手必胜则加入  $\text{redWin}(v)$ ，若  $v$  是红必胜则加入  $\emptyset$ ，若  $v$  是蓝必胜则加入  $\text{firstWin}(v)$ 。

综上所述，我们可以把  $\text{redWin}(\text{root})$  通过递归求出。

实现时我们只需要记录每个叶子是否可以成为先手必胜的首次选择即可，递归到叶子的时候打上标记。复杂度  $O(n)$ 。