

An easy problem about trees 解题报告

安徽师范大学附属中学 罗哲正

1 试题来源

MemSQL Start[c]UP 2.0 - Round 2 Problem F

链接: <http://www.codeforces.com/contest/457/problem/F>

2 试题大意

你有一棵 n 个点的正则二叉树每个叶子节点有权值 a_i , Pieguy和Piegirl轮流操作, 每次选择两个父亲相同的叶子, 把他们都删掉, 并把其中一个的值赋给它们的父亲, 直到只剩下一个根。Pieguy想最大化最后根上的值, 而Piegirl想最小化这个值, Pieguy先操作, 求最后根上的值。 T 组数据。

$T \leq 100; n \leq 250; a_i \leq 1000$.

时间限制: 2s

空间限制: 256MB

3 算法介绍

我们先考虑弱化版, a_i 只有0和1两种取值, 这样游戏就可以用输赢表示了, 例如对于Pieguy来说1就是赢。

我们考虑使用树形DP, 对 x 所在子树求解。简单的考虑可以分如下两种情况 (以下默认主观先手玩家):

1. 子树内非叶节点数是奇数, 称为odd tree
2. 子树内非叶节点数是偶数, 称为even tree

odd tree的最后一步由我们选择，而even tree的最后一步由对手选择。

从简单的开始考虑，一棵odd tree的两个孩子都是even tree，显然我们只需要赢得两棵子树中的一棵即可。

但是如果一棵odd tree的两个孩子都是odd tree，上面的办法就不管用了，因为如果我们在能赢的那棵子树中游戏，对手可以在另一棵子树中随便操作，最后一定会多出一步。于是在我们“能赢”的那棵子树中对手就有机会弃权一步，这会影响我们的决策，可能导致输赢关系发生变化。

我们来修正这个问题，在DP的时候再记录一个量表示是否有一次skip操作的机会，即一个无用的odd子树。

可以发现不用记录skip的次数，有两次skip是没意义的，如果双方都有skip的机会，在一方使用之后，另一方可立即跟着使用，可以抵消影响。

接下来我们只要对于以下六种情况，讨论DP即可。

- even-even-noskip
- even-even-canskip
- odd-even-noskip
- odd-even-skip
- odd-odd-noskip
- odd-odd-skip

其中canskip表示可以skip也可以不skip，skip表示必须skip，noskip表示不能skip。

3.1 even-even-noskip

最后一步操作是我们的，所以只要不使用skip赢得任何一个子树就可以了。

3.2 even-even-canskip

由于这里的skip操作只是相当于在某个子树中多了一步，任何一方如果不使用skip操作会输，那么一定会使用skip操作，所以类似于odd-even-noskip，最后一步是对手的。

由于我们先手可以优先使用skip操作，所以我们能赢的条件是我们能用skip操作赢得一个子树并且对手不能不通过skip操作赢得另一个子树。

3.3 odd-even-noskip

最后一步是对手的。

一种情况是：如果我们不使用一个skip操作赢得odd子树，且对手不能不使用skip操作赢得even子树，我们就能赢。

因为我们可以从odd开始操作，这样就变成了even-even-noskip的情形，这种情况下对手在两棵子树都输了，我们就赢了。

另一种情况：如果我们能使用skip操作赢得even子树，并且对手可以使用skip操作时不能赢得odd子树，我们也能赢。我们在even子树内使用skip操作就转化成了odd-odd的情况，对手可以使用其中一个子树来skip，所以我们要保证任意一个子树对手都不能使用skip赢。

3.4 odd-even-skip

最后一步是我们的。

这种情况很简单，我们只要把skip当成odd子树上的额外操作，所以若能不用skip操作赢得even子树，或者使用skip操作赢得odd子树，我们就赢了。

3.5 odd-odd-noskip

最后一步是我们的。

我们只要可以使用一个skip操作赢得任何一个子树就能赢，如果需要就把另一棵子树用来做skip操作即可。

3.6 odd-odd-skip

最后一步是对手的。

假如立即就用skip操作，如果对手输了我们赢了。如果不立即使用，如果我们能用skip赢得一棵子树，并且对手不能不使用skip操作赢得另一棵子树，我们就赢了。

a_i 为0,1的情况能做我们就能二分了,但是仔细分析我们的DP过程,其实只要把输赢信息改成最大值最小值信息统计就可以了。

时间复杂度 $O(Tn)$ 。

4 总结

这是一道需要思考很多东西的博弈题,第一步把问题简化为只有01的情况能很大程度上降低难度,把权值大小改成直观的输赢信息。而在求解输赢信息的DP中,引入skip/canskip/noskip帮助记录状态,对每一种情况进行了细致的分析。需要选手有较强的逻辑推理能力和完整处理细节的能力。这题在Codeforces上现场0AC,至今AC人数也极少。