NOIP2023 模拟赛-XI

orj

November 2023

A 二叉树

题目大意

给定一棵二叉树，m 次操作，每次交换一个节点的左右儿子。
最小化最终二叉树的叶子序列的字典序。

思路

令 $dp_{i,j}$ 表示序列为 i 子树交换 j 次的最小字典序序列，类似树上背包维护 dp。
即 $dp_{u,i} \leftarrow \{dp_{ls,a} + dp_{rs,i-a}, dp_{rs,a} + dp_{ls,i-a-1}\}$
($+$ 表示字符串拼接)。
dp 转移需要序列拼接 / 比较字典序，暴力处理复杂度为 $O(n^3)$。

Question:
直接存储序列需要 $O(n^3)$ 的空间，无法通过。

Answer:

1. 每次转移后子树的信息可以被释放，可以用 vector 维护释放。
2. 可以预先分配空间，每次转移后，覆盖儿子用过的部分。
3. 用类似可持久化的结构维护，每次拼接操作新生成一个节点。

A 二叉树

Tips:

1. $n = 1000$ $O(n^3)$ 看起来有些吓人，但是估计复杂度时应该取实际上有效的 $n' = n/2 = 500$。
2. 由于可以操作一个节点多次，所以实际上 $m-2, m-4, \cdots$ 次操作也都可以
3. 还需要特判没有可操作点的情况。

A 二叉树

使用 (3) 可以规避暴力拼接序列的操作，且有进一步优化的空间。

A 二叉树

总结

- 题目要点：dp / 优化。
- 预估难度：简单/中等。
- 得分情况：0/100。
- 丢分分析：不判 -1 导致的。

B 排列

题目大意

设函数 $f(p)$ 为通过相邻交换操作实现 next_permutation。
求将排列 $p = \{1, 2, \cdots, n\}$ 变为 q 需要:
(1) 调用多少次 f
(2) 执行多少次相邻交换。

B 排列

Q1: Cantor 展开。

B 排列

Q2: 计算相邻交换数

考虑类似 Cantor 展开的思路，先计算长度为 n 的排列从 $1\sim n$ 变为 $n\sim 1$ 的操作数，记作 F_n。
考虑递推过程：

$$
\begin{aligned}
F_n =&[1\sim n]\to[n\sim 1]\\
=&[1,2\sim n]\to[1,n\sim 2]\to[2,1,3\sim n]\\
&\to\cdots\\
&\to[i,1\sim i-1,i+1\sim n]\to[i,n\sim i+1,i-1\sim 1]\\
&\to[i+1,1\sim i,i+2\sim n]\to\cdots\to[n\sim 1]\\
=&nF_{n-1}+\sum_{i=1}^{n-1}[i,n\sim i+1,i-1\sim 1]\to[i+1,1\sim i,i+2\sim n]
\end{aligned}
$$

B 排列

通过模拟可以发现：

$$[i, n \sim i+1, i-1 \sim 1] \to [i+1, 1 \sim i, i+2 \sim n] = \binom{n-2}{2} + (n-1)$$

模拟过程如下：

1. 将 $i+1$ 移动到最开头，需要 $n-i$ 步
2. 随后将 $[n \sim i+2, i-1 \sim 1]$ 的子列完全翻转，需要 $\binom{n}{i}$ 步
3. 将 i 从第二位移动到 $1 \sim i-1$ 后面，需要 $i-1$ 步。

故 $F_n = nF_{n-1} + (n-1)(\binom{n-2}{2} + (n-1))$。
用类似的式子替换 Cantor 展开的操作即可。

B 排列

总结

- 题目要点：推性质计数 / BIT。
- 预估难度：中等。
- 得分情况：正常分布。
- 丢分分析：无。

C GAME

题目大意

有 $2n+1$ 个球，每个球上有一个数字，一开始你选择一个球取出。
随后进行如下游戏 n 轮：

1. A 取出一个球 x_i。
2. B 取出一个球 y_i。

最终得分为 $\max\{x_i \operatorname{xor} y_i\}$。
A 希望最大化得分，B 反之。
对于 $2n+1$ 种一开始被取出的球，输出最终的得分。

C GAME

Sub: $2n$ 个球的游戏。
B 可以预先为所有球配对，每次 A 选择时就选择对应的一个。
问题转变为配对最小化最大的异或值。

对于所有的值建立 trie 树，则有如下观察：

1. 若左右子树均有偶数个值，则最优策略一定是分别在子树内做匹配。
2. 否则，左右子树个数均为奇数，此时最优解一定只选择跨子树的一对，且这一对树的异或值决定答案。

C GAME

$$
\begin{aligned}
&\text{Function } calc(node)\\
&\quad \text{if } node = null\\
&\quad\quad return\ 0\\
&\quad \text{assert } node \to count\ is\ even\\
&\quad \text{if } node \to lson \to count\ is\ even\\
&\quad\quad return\ max(calc(node \to lson, node \to rson))\\
&\quad else\\
&\quad\quad return \min_{x \in node \to lson} \{query(node \to rson, x)\}
\end{aligned}
$$

其中 query 函数复杂度为 $O(\log a)$，至多被调用 $O(n)$，复杂度为 $O(n \log a)$。

C GAME

将该算法拓展到 $2n+1$ 个球，记作 $calc2$。
不妨设左儿子有奇数个数，右儿子有偶数个数。

1. 若一开始选出的数在左儿子，则递归进入左儿子 $calc2$，得到的结果与右儿子 ($calc$) 的答案取 $\max$。
2. 若一开始选的数在右儿子，则答案由跨过子树的一对值决定。

C GAME

$$
\begin{aligned}
&\text{Function } calc2(node)\\
&\quad \text{if } node = null\\
&\quad\quad return\ []\\
&\quad \text{assert } node \to count\ is\ odd\\
&\quad \text{if } node \to lson \to count\ is\ even\\
&\quad\quad swap(node \to lson, node \to rson)\\
&\quad lans = tomax(calc2(node \to lson), calc(node \to rson))\\
&\quad rans = [\min_{y \in node \to rson, y \neq x} \{query(node \to lson, y)\} \big| x \in node \to rson]\\
&\quad return\ lans + rans
\end{aligned}
$$

rans 求解只需要求最小值次小值，复杂度有粗略的上界 $O(n \log^2 a)$

总结

- 题目要点：trie 树 / 贪心。
- 预估难度：中等。
- 得分情况：大部分人没看出博弈本质，甚至懒得打 2^n。
- 丢分分析：不懂。

D 雪

题目大意

给定序列 a_i, b_i，维护操作：

1. 对于 b_i 区间加。
2. 按照 a_i，从左到右建立单调栈，计算 v 对应单调栈上所有 b_i 之和。

D 雪

Sub: $l = r$

对于单调栈建树，转化为单点加，链查询。
进一步转化为 BIT 维护 dfs 序上区间加、单点查。
复杂度 $O(n \log n)$。

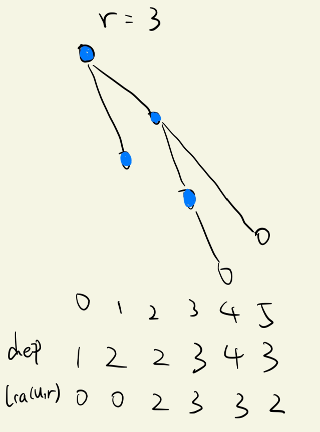
D 雪

Sub: a_i 随机生成

对于单调栈建树，维护区间加 $+$ 暴力跳链。

D 雪

加入根节点 0，考虑 $[0, r]$ 加法对于单调栈树上 u 的祖先链的贡献次数。



1. 若 $u \le r$，则贡献次数为 dep_u。
2. 若 $u > r$，则贡献次数为 $dep_{lca(u,r)}$。

解法 1
将区间操作和链操作拆成 +1,-1 两个。
按照时间分治，归并所有操作对应的 u, r。
对于 $u \leq r$ 的情况直接统计，对于 $u > r$ 的情况，所有操作建立虚树，通过在虚树上做 dp 进行统计。
使用 st 表维护虚树，复杂度为 $O(n \log n)$。