登山 题解

xiaolilsq

2024 年 7 月 20 日

题意简述

给定一棵以 1 为根的外向树，其中除 1 以外的每个节点 i 额外向它的 l_i 到 r_i 级父亲连边，用 dp_i 表示节点 i 的深度。
问对于每个点 x，有多少条任意长度的路径 $x = x_0, x_1, \ldots, x_m = 1$ 满足对于任意 $1 \le i \le m$：要么有 x_i 是 x_{i-1} 的祖先，且 $l_{x_{i-1}} \le dp_{x_{i-1}} - dp_{x_i} \le r_{x_{i-1}}$，且对于任意 $0 \le j < i$，满足 $dp_{x_i} < dp_{x_j} - h_{x_j}$；要么有 $p_{x_i} = x_{i-1}$。
T 组数据，保证 $1 \le T \le 4$，$2 \le n \le 5 \times 10^4$，$2 \le \sum n \le 2 \times 10^5$，$1 \le p_i < i$，$1 \le l_i \le r_i \le dp_i$，$0 \le h_i < dp_i$。

算法一

我会爆搜！
按照题目要求搜索路径，然后按照定义判断路径是否合法即可。
期望得分：5。

算法二

我会记忆化搜索！

对于任意合法路径的一个前缀 $x_0, x_1, \ldots, x_k$，容易发现我们只关心 x_k 以及前面所有 $dp_{x_i} - h_{x_i}$ 的最小值，就能够确定后续哪些走法是合法的。

所以我们可以设 $f_{i,j}$ 表示从 i 出发，且下一次往上跳的时候至少要跳到深度小于 j，到 1 的方案数。

状态数 $O(n^2)$，直接转移复杂度 $O(n)$，时间复杂度 $O(n^3)$。

期望得分：15。

算法三

我会观察题目性质！

注意到题面中有一句“由于上述限制每次往上冲刺都会到达比上一次冲刺更近的位置”，这启发我们将按照每次向上跳的地方将路径分割。

对于路径 $x_0, x_1, \ldots, x_k$，如果 x_{k-1} 到 x_k 是往上跳，那么肯定有 $dp_{x_k} < dp_{x_i} - h_{x_i}$ 对任意 $0 \le i < k$ 成立，注意到 $h_{x_k} \ge 0$，所以 $dp_{x_k} - h_{x_k} < dp_{x_i} - h_{x_i}$，由此我们得知 x_k 这一个点的限制已经比之前所有点的限制要更强了。

于是我们直接设 f_i 表示 i 到 1 的合法路径，然后直接考虑下一次往上跳会跳到哪里，当然下一次往上跳必然跳到 i 的祖先。用 $p_{i,j}$ 表示 i 的 j 级父亲，而 $c_{i,j}$ 表示 i 下到子树内部某个点然后往上跳到 i 的 j 级父亲的方案数，那么我们有：

$$f_1 = 1, f_i = \sum_{j=1}^{dp_i} c_{i,j} f_{p_{i,j}}$$

算法三

求 $c_{i,j}$ 也很简单，因为往下跳的时候没有任何限制，直接看往哪个儿子跳：

$$c_{i,j}=\begin{cases}0 & 1\le j\le h_i\\ [l_i\le j\le r_i]+\sum_{p_k=i}c_{k,j+1} & h_i<j\le dp_i\end{cases}$$

时间复杂度 $O(n^2)$。
期望得分：25。

算法四

我会线段树合并！
实际上看到 $c_{i,j}$ 转移方程，对类似结构较为熟悉的选手应该立马就会想到可以使用线段树合并维护 c 的求解，c_i 其实就是将孩子的所有线段树合并，然后在 $[l_i, r_i]$ 段加 1，最后再截断一个前缀/后缀而已，由此如果我们只需要维护 $c_{i,j}$，这是非常容易做到的。
但是问题在于我们要求的是 f_j 与 $c_{i,j}$ 乘积的和，而我们线段树合并到 i 这个节点时 i 祖先的 f 值还没有被算出来，怎么办？
线段树合并是从下往上，求 f 值是从上往下，我们直接大胆一点，考虑线段树合并的逆过程，由于所有操作都是线段树合并的逆，而线段树合并的时间复杂度是正确的，所以这个逆过程时间复杂度也是正确的。
时间复杂度 $O(n\log n)$。
期望得分：100。

吐槽环节