pION 赛拟模题解

2023 年 9 月 27 日

括号 (bracket)

一个区间合法的条件是：

- (? 当成 1，) 当成 -1，所有前缀和都 ≥ 0；
-)? 当成 1，(当成 -1，所有后缀和都 ≥ 0。

dp f_i 表示以 i 结尾的答案，$f_r = \max(f_l + a(r-l) + b)$，区间 $(l, r]$ 合法。第一条限制相当于对一个 l，合法的 r 在一个前缀里，第二条限制相当于对一个 r，合法的 l 在一个后缀里。可以用线段树单点删除和区间查询最大值实现。

跳跃 (jump)

本题其实是 CF1495F 的加强版。

首先，对每个 $i \in [1, n]$ 求出 $j > i$ 且 $v_j > v_i$ 的最小的 j，记作 p_i。可以使用单调栈求出 p。

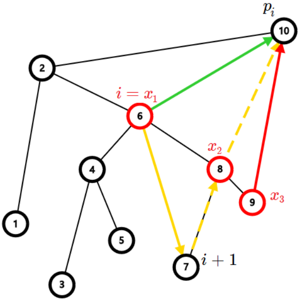
可以发现，从点 i 出发，若想要走到点 $j > p_i$，则必定要经过 p_i。证明很简单：若从 i 走到 $j > p_i$ 且不经过 p_i，则设跨过 p_i 的一步为 $x \to y$，则显然必须有 $p_x = y$。由于 $p_i > x$，故由 p_i 定义知 $v_x < v_{p_i}$，这与 $x < p_i$、$y = p_x > p_i$ 矛盾。

由此也可以得到，若 $j = p_{p_{\dots p_i}}$（至少 0 个 p，下面若满足此式，称 j 为 i 的「p 祖先」），则若想要从 i 走到 j 后面的点，则必须经过 j。i 的 p 祖先其实也就是 $v_{i \dots n+1}$ 的所有前缀最大值。

跳跃 (jump)

于是设 d_i 表示 i 走到 p_i 的最短路，考虑求出 d_i。考虑走到 p_i 的最后一步，只有做单调栈时被 p_i 弹出的点才能走到 p_i。这些点在原序列的笛卡尔树上，形如从 p_i 的左儿子向右延伸的链的一段后缀，设为 $x_1 = i, x_2, x_3, \ldots, x_k$。

i 可以一步走到 p_i，也可以走到 $i+1$。可以发现，x_2 为 $i+1$ 的 p 祖先，显然它是 $v_{i+1\ldots n+1}$ 的前缀最大值之一。于是若第一步走 $i+1$，则必定经过 x_2。



跳跃 (jump)

经过 x_2 后从 x_2 走到 p_i 的最短路可以直接调用 d_{x_2} 得到。现在需要求出 $i+1$ 到 x_2 的路径长度，由于 x_2 为 $i+1$ 的 p 祖先，从 $i+1$ 不断跳 p 直到 x_2 并把答案对经过的 d 取 $\max$ 即可，为了保证时间复杂度需要使用倍增优化。

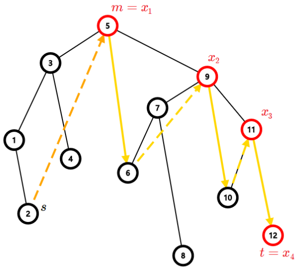
于是可以用 $d_{i+1 \dots x_2}$ 求出 d_i。可以在单调栈的过程中求出所有 d_i（每次一个点 i 被弹出时求 d_i 即可）。值得一提的是，这里需要在线地更新 p、d 并维护倍增数组。可以对每个点 i 与二进制位 k 维护 $S_{i,k}$ 为跳 2^k 次 p 跳到 i 的点构成的集合，更新 d_i 时需要更新 $S_{i,0}$ 中的 2^1 步结果，然后对于 $S_{i,0}$ 中的每个位置 j 更新 $S_{i,1}$ 中的 2^2 步结果，不断递归更新即可。

得到了 d_i，考虑求任意两点 s、t 间的距离。为了保证复杂度，可以在加入点 t 后立即求所有 (s,t) 的答案，这保证 t 为当前笛卡尔树最右链的链底。

跳跃 (jump)

设区间 $[s,t]$ 内 v 的最大值为 v_m，其为 s 的 p 祖先，于是可以先倍增跳到 m（一个方便的实现是，m 是 s 不断跳 p 而不跳过 y，跳到的最后一个位置）。由于 t 是最后加入的数，容易证明 m 也在当前笛卡尔树的最右链上，设 m 到 t 的路径依次为 $x_1=m, x_2, x_3, \ldots, x_k=t$。

若 $t \neq m$，则此时可以发现再跳 p 就会超过 t，于是只能跳到 $m+1$。容易证明 x_2 为 $m+1$ 的 p 祖先，于是可以倍增跳到 x_2；然后又只能跳到 x_2+1，然后倍增到 x_3……依此类推，最后跳到 $x_k=t$。



跳跃 (jump)

但是这样时间复杂度不对。可以对于最右链上的每个点 i，处理出最右链中前一个点到它的最短路 cd_i，于是要求的就是一段后缀的 cd 最大值。每次加入一个点时可以倍增求出 cd，需要在最右链末端加删点，维护后缀最大值，可以使用线段树实现。

总时间复杂度为 $O(n \log n)$。可以用可持久化线段树维护 cd 做到在线。

跳跃 (jump)

这里是另一个较好实现的做法。

容易发现，在单调栈的过程中，一个点 u 在最右链上的前一个点只会出现一个，即 u 左边且比 v_u 大的最后一个点，设为 cp_u（对应的 cd_u 也只有一个）。所以预处理 cp、cd，则 m 右边的答案可以通过由 t 不断跳 cp 直到 m，把答案与沿途经过的 cd 取 $\max$ 得到。于是也可以使用倍增求解。
总时间复杂度为 $O(n\log n)$，支持在线。

集了个合 (set)

对于 $j\in\{1,\cdots,n\}$，设 $T_j:=\{i\in\{1,\cdots,m\}:j\in S_i\}$。那么合法的条件等价于任意 T_j 非空且对于任意 $j\neq k$ 有 T_j,T_k 互不包含。

那么相当于，我们要在 $2^{\{1,\cdots,m\}}$ 中选择以“集合包含”为偏序关系的最长反链。

我们断言，最长反链就是 $\{1,\cdots,m\}$ 的大小为 $\lceil m/2\rceil$ 的子集组成的集合，即 $n=\binom{m}{\lceil m/2\rceil}$。

为证明这一点，根据 Dilworth 定理，考虑找出 $2^{\{1,\cdots,m\}}$ 的一组链覆盖，使得链的数量也为 $\binom{m}{\lceil m/2\rceil}$。

考虑构造单射 $f_{m,i}$，其中 $1\leq i\leq\lceil m/2\rceil$，它是从 $\{1,\cdots,m\}$ 的大小为 i 的子集到 $\{1,\cdots,m\}$ 的大小为 $i-1$ 的子集的单射。构造完后，容易找出一组满足要求的链覆盖。

集了个合 (set)

使用归纳构造：

对于 $i \le \lceil (m-1)/2 \rceil$ 的 $f_{m,i}(T)$，若 T 不包含 m，则令 $f_{m,i}(T) := f_{m-1,i}(T)$；若 T 包含 m，则令 $f_{m,i}(T) := f_{m-1,i-1}(T \setminus \{m\}) \cup \{m\}$。

对于 m 为奇数且 $i = \lceil m/2 \rceil$，构造 $f_{m,i}(T) := f_{m,i-1}(\{1, \cdots, m\} \setminus T)$。

数树 (tree)

Subtask 1

枚举每个重心考虑它的贡献。先枚举它所在连通块的大小 sz，然后它所有儿子子树的大小不超过 $\frac{sz}{2}$，每个 sz 对所有儿子做一遍背包，复杂度 $O(n^4)$。

数树 (tree)

Subtask 2

容斥，一个点不合法当且仅当它一个儿子子树大小超过 $\frac{sz}{2}$，而这样的儿子至多一个。枚举是哪个儿子，然后做一遍其他儿子的背包，具体是先做出前后缀的背包，然后拼起来。复杂度 $O(n^3)$。

进一步研究点 u 不合法的条件：存在点 v 与 u 相邻，断掉 (u, v) 后 u 所在连通块大小比 v 所在连通块大小要小。相当于我们要求出两个东西：$f_{u,i}$ 表示以 u 为根的子树，连通块大小为 i 的方案数，$g_{u,i}$ 表示删掉以 u 为根的子树，fa_u 所在连通块大小为 i 的方案数。

$f_{u,i}$ 是经典的树上背包，而 g_u 可以从 g_{fa_u} 转移过来，拼上 g_{fa_u} 其他儿子的 f，也可以前缀拼后缀。用 FFT 转移，复杂度 $O(n^2 \log n)$。

注意到 $g_{u,i}$ 的 i 可以跟 siz_u+1 取 $\min$，这样对于一个 u 我们可以把它的复杂度做到 $O(siz_u(siz_{fa_u}-siz_u))$，跟树上背包一致，复杂度是 $O(n^2)$。