

绘制二叉树(binary)

按照题意模拟即可。

魔法师(magic)

首先将序列分割成 $[1, \frac{n}{2}]$ 和 $[\frac{n}{2} + 1, n]$, 考虑某一个序列的一个数 p_x , 如果它不是前缀最大值, 则它必然和它的到前面距离它最近的前缀最大值 p_i 的这一段在一起处理。

证明考虑对于 p_i 如果被放进序列, 说明另一堆有 $t > p_i$, 由于 p_i 时前缀最大值, 但是 $p_{i+1} \sim p_x$ 不是, 则显然有 $t > p_i > p_j, j = i + 1, i + 2 \dots x$ 。

这也就意味着, 合并的过程可以看作将两个序列按照前缀最大值的位置开头分割成若干区间, 然后以这些区间的第一个数的权值进行归并排序。

如果排序之后, $\frac{n}{2}$ 和 $\frac{n}{2} + 1$ 之间刚好是分割开的, 则进行一次分割不会影响这些区间的数量和结构, 也就意味着序列不变。

由于每一次序列变化意味着增加了一个区间, 所以有效的变换只有 $O(n)$ 次。

暴力维护或者使用平衡树维护区间, 用线段树上二分快速求出区间第一个 $> k$ 的数, 即可维护这个过程。具体的:

首先进行第一次操作的初始化, 然后将所有区间按照 $\frac{n}{2}$ 的位置分裂, 取出包含了 $\frac{n}{2}$ 和 $\frac{n}{2} + 1$ 的区间 (如果没有就意味着序列不会再变化), 将其分成 $\frac{n}{2}$ 及以前的这个区间, $\frac{n}{2}$ 以后的区间按照前缀最大值分割成的若干区间, 将其按照第一个数的权值插入原区间即可。

时间复杂度 $O(N \log N + Q \log N)$ 。

环上排序信息最优分割(ring)

将 n 个序列拼成一个环, 就是相当于要在环上的 n 段中各断开一个未知, 然后取计算代价。不妨即第 i 各切断的位置 x_i 需要在区间 $[l_i, r_i]$ 中。也就有 $l_i = \sum_{j=1}^{i-1} m_j + 1, r_i = \sum_{j=1}^i m_j$ 。

环的问题并不好处理, 所以考虑断环为链, 枚举第一个区间的 x_i , 进行如下的 DP, 即 f_j 表示某一个分割位置在 j (由于所有 $[l_i, r_i]$ 区间不交, j 是第几个分割点是唯一确定的), 那么对于 $x \in [l_i, r_i]$, 有转移 $f_x = \min_{x' \in [l_{i-1}, r_{i-1}]} (f_{x'} + val(x', x - 1))$, 其中 $val(l, r)$ 表示环上 $[l, r]$ 元素构成的序列的代价。

发现代价不好单个快速查询, 但是是能够支持快速的指针移动的: 如果能够维护出 $[l, r]$ 排序后的结果, 可以 $O(\log n)$ 的加入/删除一个元素。因此对于单个 x_i 暴力 DP 的时间复杂度可以做到 $O(Mm \log m)$ 。

容易证明, val 函数满足四边形不等式, 从而说明 DP 转移具有决策单调性, 因此可以使用分治优化 DP 转移。单次时间复杂度 $O(M \log^2 m)$ 。

更进一步的, 如果对于某一个 x_1 , 其对应的最优决策点为 x_2, x_3, \dots, x_n 。可以证明 $x_2, x_3 \dots x_n$ 是关于 x_1 单调的。

因此, 可以套用一个类似于分治优化的结构。对于 $x_1 \in [L_1, R_1]$ 的答案, 知道对于 $x_i \in [L_i, R_i]$ 。

通过 DP 计算出 $x_1 = \left\lfloor \frac{L_1 + R_1}{2} \right\rfloor$ 的最优转移点 $x_2, x_3 \dots x_n$ 。那么就可以说明, 对于

$x'_1 \in [L_1, x_1 - 1]$ 的答案, 必然有转移点 $x'_i \in [L_i, x_i]$; 对于 $x'_1 \in [x_1 + 1, R_1]$ 同理。

为了保证复杂度的正确性，不妨让 N 套用这个分治结构，可以证明当前的时间复杂度为 $O(M \log^3 M)$ 。其中一个 $O(\log M)$ 来自维护 val ，另外两只 $O(\log M)$ 分别来自两个分治算法。

注意到在维护 val 的时候，如果只需要指出删除和撤销，就可以通过提前排序，然后用链表维护来做到 $O(1)$ 移动指针，这样可以将复杂度优化到 $O(M \log^2 M)$ 。

野猪(wildboar)

根据贪心，我们必然会选择最短路作为选择的路径。

但是由于不能走“回头路”，所以我们还需要找到起点和终点均与最短路不同的“次短路”。发现仍然不够，我们还需要起点和最短路不同，终点和次短路不同；起点和次短路不同，终点和最短路不同：这样的两条路径。

可以证明存在最优解必然在有这 4 种路径组合得到。

考虑如何维护快速的合并转移，对于每一个 A_{i-1}, A_i, A_{i+1} ，我们可以得到一个矩阵表示 A_{i-1} 到 A_i 的四条路径和 A_i 到 A_{i+1} 的四条路径之间分别是否可以转移。发现修改和查询类似 DDP，使用矩阵维护 $\min, +$ 矩阵乘法即可。

可以使用线段树维护，时间复杂度 $O(m^2 \log m + n^2 + L + T \log L)$ 。