NOI 2016天津市代表队选拔赛 day1题解

Contents

A tree

对于70%数据的做法:

注意到一个点的标记只会对这个点所在的子树有影响，所以我们考虑将树的问题转成对应的DFS序列的问题。

那么问题就转变为：给定一个序列，支持两种操作：1. 将某个区间修改为某个值 2. 查询某一点的值 利用线段树可以很简单的做到。

时间复杂度为$O(n + qlogn)$

对于100%数据的做法： 考虑离线处理。

我们将所有的操作倒序处理，利用并查集进行维护。对于一个修改操作（即失去标记），相当于将这一联通块块与它父亲的联通块合并。对于一个查询操作，直接查询所在的块的父亲即可。

实现中有些细节需要注意，比如每个点可能会被标记多次，所以要以最早的被标记时间做为“失去标记”的时间。

时间复杂度为$O(n + q\alpha(n))$

B sort

对于30%的数据，只要模拟题目要求暴力求解即可，复杂度为$O(nmlogn)$。

对于100%的数据，可以通过二分答案，每次二分的值为a，将整个序列中大于等于a的值变为1，小于a的值变成0。这样序列中只有0和1，对于每一次部分排序，我们可以通过线段树的区间更新，若为升序排序，则把该区间的0全部排在区间前段，1排在区间后段，最后全部部分排序结束后，检验位置p上的值是1还是0，来继续调整下一次二分。直到二分结束后即可。复杂度为$O(mlog^2n)$。

C seq

对于20%的数据，可能的情况至多只有101种，直接暴力dp，dp[i]表示以第i个元素结尾的最优结果。第i个元素可以放在第j个元素后边，当且仅当对于任意一种情况均有$a[j] \le a[i]$。 – 复杂度$O(n^2m)$

对于50%的数据，观察可发现我们并不用记录所有的情况。对于第i个元素，只需要记录其原始值，变化的最小值，变化的最大值。

对于第i个元素，记其原始值为a[i]，变化的最小值为b[i]，变化的最大值为c[i]

第i个元素可以放在第j个元素的前边(i¡j)，当且仅当

1. $a[i] \le a[j]$

2. $c[i] \le a[j]$

3. $a[i] \le b[j]$

– 复杂度$O(n^2)$

对于100%的数据，继续沿用上一个思想，注意到i能否放在j的后边，仅与j位置元素的原始值和变化后的最大值有关。所以这道题和求二元组的最长不降子序列相似。并且元素可以连接放这一关系具有传递性，所以可以二分答案。可以得到多种解法。

1. 将dp[i]的值记录在二维平面点(a[i], c[i])上，那么dp[i]的值就是在点(min(a[i], b[i]), a[i])左下方的所有点（含边界）上的权值的最大值+1，这个点修改和区间查询可以通过kd树来完成。 – 复杂度$O(n^{1/2})$，写的不好可能会被卡常数

2. 存放n+1个桶，将(a[i], c[i])放入第dp[i]个桶内。由于关系具有传递性，即i能放在j前面，j能放在k前面，则i能放在k前面。所以计算dp[i]时可以二分答案，然后在对应的桶内查找有没有可以放在i前面的元素。桶内查找操作可以通过一个splay或者动态建树的线段树完成。 – 复杂度$O(nlog^2n)$

3. 使用整体二分的思想，若要求出dp[1..n]，令t = (1 + n) / 2，先处理出dp[1..t]的所有值，然后处理dp[1..t]向dp[t+1..n]的转移，再处理dp[t+1..n]内部的转移。处理dp[l..t]向dp[t+1..r]的转移需要O(klogk)的复杂度，其中k=r-l+1 – 复杂度$O(nlog^2n)$