ION 1202 模拟赛 题解

ASDFZ

1202 年 -1 月 -i 日

七管荧光灯（qgygd）

Author: Suiseiseki

首先我们打表可以发现，先手必败的状态是 $1,2,3$ 号管上的数相等，$5,6,7$ 号管上的数相等，并且 1 号管上的数，4 号管上的数和 5 号管上的数异或和为 0。

证明十分容易，我们先考虑游戏的终止状态，也就是所有的管子上的数都是 0，显然异或和为 0，必败态显然无法转移至必败态。

接下来考虑证明必胜态一定能转移到必败态，我们取出 $1,2,3$ 号管中最小值所在的管子，记为 x，$5,6,7$ 号管中最小值所在的管子，记为 y，先将 $x,y,4$ 这些管子放在一边，剩下的管子全选，然后再选 $x,y,4$ 中值最大的，容易证明不会成环，显然可以通过调整这些管子使其到达必败态。

在知道这个结论之后，我们可以用数位 dp 求出必败态的种类数，简单来说，就是先容斥为求只有最大值限制的方案数，然后在设 $f_{i,0/1,0/1,0/1}$ 表示当前 dp 到了第 i 位，第 $1,2,3$ 号管是否达到上界，4 号管是否达到上界，$5,6,7$ 号管是否达到上界，时间复杂度 $O(\log n)$。

混乱（chaos）

Author: Seniorious

不难想到按照 k 的大小分治，最朴素的想法是 k 小的时候直接树套树求出最小的覆盖某个区间的技能，单次 $\mathrm{O}(k^2\log^2 n)$，k 大时维护 f_i 代表覆盖了 $S_1\sim S_i$ 的最小代价，转移是考虑修改 (l,r,w) 其中 $r\in[S_i,S_{i+1})$，$f_i\leftarrow\min\limits_{j\geq l}f_j+w$ 这个可以用树状数组单次 $\mathrm{O}(n\log n)$。

先考虑 k 大的部分，发现我们的修改次数远小于询问，我们可以用分块代替树状数组，这样询问就是 $\mathrm{O}(1)$ 的。

再考虑 k 小的部分，简单的一个优化是将操作分块，将树套树改为主席树，定期重构主席树，每次询问对散块的修改随便处理一下。

到这里为止都没有用到 l,r 随机的性质，我们考虑这有什么用：如果对序列分块，那么以某一块中的点为左/右端点的修改是不多的。令 $\mathrm{dis}(i,j)$ 为覆盖 $[S_i,S_j]$ 的最小修改。考虑分为三部分：以 S_i 所在块为左端点的修改，以 S_j 所在块为右端点的修改，左端点为 S_i 左边的整块右端点为 S_j 右边的整块。前两种做法相同，下面只讨论第一种：可以维护出以每个块为左端点的修改（按右端点排序，排序可以修改时插入排序），将 $S_j(j\geq i)$ 和右端点一起做双指针，维护最小值。第三种我们可以维护左端点在第 i 个块中，右端点所在的块大于等于 j 的最小值，一样固定左端点，对右端点做一次双指针即可。

通过两个部分的不同做法的组合，你可以获得 $\mathrm{O}(n^{1.5}\log^{1.5}n)$ 到 $\mathrm{O}(n^{1.5})$ 的各种复杂度。那么这题如果没有 l,r 随机是否能做到 $\mathrm{O}(n\sqrt{n})$ 呢？答案是能的，只要每 $\sqrt{n}$ 次重新分一次块，重构即可，但是常数实在太大，被 $\mathrm{O}(n\sqrt{n\log n})$ 吊着打（也可能是验题人实现不够优秀），所以这题加上了特殊限制。

超立方体（hypercube）

Author: Suiseiseki

我们先约定一些记号：

- 记 $\mathrm{dis}(u,v)$ 为 u 到 v 的异或和
- 记 dep_u 为根到 u 的异或和
- 记 $\oplus$ 为异或运算

接下来的分析我们都假定 1 是整棵树的根。

我们考虑如何求出两个点的 LCA，假设这两个点是 u,v，那么这两个点的 LCA 就是 $dep_u \oplus dep_v \oplus dis(u,v)$，我们可以据此得出一个询问次数为 $O(n^2)$ 的做法，并且还可以根据「JOISC 2019 聚会」该题的思路获得度数较小的部分分。

接下来有两种考虑方法，一种是采用点分治（其中一位验题人使用的就是这种方法），但是由于其常数较大，如果不精细实现难以通过，所以此处略去。

另一种方法是考虑重链剖分，我们先将当前已知的点建虚树并对其轻重链剖分，接下来，假设我们在点 r，我们要加入的点是 u，考虑如何操作（先通过 $n-1$ 次询问预处理出 dep_u）。

首先显然可以通过一次询问判断当前点是否在重儿子中，若在重儿子所在链上，则可以通过二分来确定其位置。

若点 u 不在点 r 的重儿子中，则我们希望能够通过一次询问得到其在哪一个轻儿子所在子树中或者是用 $\log n$ 次的询问将其加入 r 的轻儿子集合。

令轻儿子的个数是偶数（这可以通过在轻儿子个数是奇数时将重儿子加入轻儿子集合做到），接下来我们查询 u 和轻儿子集合，假设返回 v'，我们再记 $s=\sum_{v\in son_r} dep_v \oplus dep_u$（此处的 son_r 表示 r 的轻儿子集合），令 $v=v'\oplus s$，接下来可能会出现一下几种情况：

- $v\in son_r$，那么点 u 在 v 的子树中，递归下去处理即可。
- $v=0$，那么点 u 是 r 的轻儿子，将其挂在 r 下即可。
- $v=t\oplus r$，证明 t 是 r 的轻儿子，且 t 的子树中含有 u 和 son_r 中的一个点，可以通过二分找到 t 包含 son_r 中的哪一个点。

容易证明不会出现 v 重复的情况（如果您认为会出现重复，将您认为会重复两种状态异或一下之后就会明白为什么这两个数不相同了）。

每一次重新进行重链剖分（请不要使用随机剖分，该做法不知道为什么会有一定的概率超过询问次数限制），时间复杂度为 $O(n^2\log n)$，询问次数为 $O(n\log n)$，粗略实现的标程只需要最多不超过 37000 次询问。