NOIP 2019 模拟赛 Contest 1

diamond_duke

题目名称	序列	灯泡	比赛
可执行文件名	`seq`	`bulb`	`match`
输入文件名	标准输入	标准输入	标准输入
输出文件名	标准输出	标准输出	标准输出
时间限制	1s	1s	1s
内存限制	512MB	512MB	512MB
子任务个数	5	5	5
题目类型	传统型	传统型	传统型

请注意： 评测时开启 O2 优化和 C++11 编译选项，栈空间限制同空间限制。

1 序列

奇数和偶数显然是独立的，我们只考虑其中一种即可。

如果没有要求字典序最小的话，则显然相对位置不变的方案是最优的，那么我们可以直接得到一种合法方案以及最小代价。

我们用 x_i 表示第 i 个数是往左，往右还是不变，那么按 x_i 分段后显然每一段是独立的，否则代价一定大了。

那么我们考虑 x_i 相同的一段。如果他们都是向左的，那么我们可以按照从大到小的顺序，每个数字都尽可能向后面放。而如果都是向右的，我们按照从小到大的顺序每个数字都尽可能往前面放即可。可以发现，这样的两种放法都可以最小化字典序，因此都是对的。

时间复杂度：$\Theta(n \log_2 n)$。

2 灯泡

考虑建出一张图，他的点就是原来的灯泡 $1, 2, \cdots, n$。然后如果 i 和 $i+1$ 都是亮着的，那么就把他们之间连一条边。

那么一个极长亮灯区间就对应图中的一个联通块。因为链也是树，所以连通块数等于点数减边数，即极长亮灯区间数等于亮着的灯泡数减去连续亮着的灯泡数。

前者显然可以非常方便地维护，考虑如何维护这个连续亮着的灯泡数。

设阈值 B，则我们可以把所有颜色按照对应灯泡个数和 B 的关系分为大小两种。

若翻转的是小的颜色，则我们直接可以暴力枚举这种颜色中的所有点，然后计算连续亮着的灯泡数。

否则如果翻转的是大的颜色，则和它相邻的颜色有两种：小的和大的：

- 如果是大的，那么因为大的颜色只有 $\frac{n}{B}$ 种，所以只要先预处理出任意两种大的颜色之间有多少条边可以连，然后直接枚举这个另外的大的颜色即可。
- 对于小的的情况，我们考虑在小的那里处理。即，在枚举小的点的时候，如果周围遇到了一个大的颜色，则我们就在这个大的颜色上打一个标记，表示如果它翻转了那么会造成多大的改变即可。

时间复杂度：$\Theta(q(B + \frac{n}{B}))$，取 $B = \sqrt{n}$ 即可做到时间复杂度 $\Theta(q\sqrt{n})$。

3 比赛

显然对于某个 k，若存在这样的 k 个人，那么必定唯一。

这是因为我们如果存在某两个合法集合 S 以及 T，一定存在 $u \in S$ 而 $u \notin T$ 以及 $v \in T$ 而 $v \notin S$。根据合法集合的定义，我们从 S 的定义可得 u 胜过 v，而从 T 的定义可得 v 胜过 u，矛盾。因此这样的集合一定唯一。

设 $F_{n,k}$ 表示 n 个人中存在这样 k 个人的概率，考虑转移。

考虑如何计算 $F_{n+1,k}$，如果 $n+1$ 在集合外，那么他一定要输给前面的这 k 个人，而这 k 个人的编号都比他小，因此概率为 p^k。如果他在集合内，那么他要赢前面的 $n-k+1$ 个人，因此概率为 q^{n-k+1}，其中 $q=1-p$。因此，我们有

$$F_{n+1,k} = F_{n,k} \cdot p^k + F_{n,k-1} \cdot q^{n-k+1}$$

同时，我们也可以通过考虑 1 这个人来做出转移。如果 1 在集合外，那么他要输给后面的这 k 个人，而这 k 个人的编号都比他大，因此概率为 q^k。如果他在集合内，那么他要赢后面的 $n-k+1$ 个人，因此概率为 p^{n-k+1}。因此，我们又有

$$F_{n+1,k} = F_{n,k} \cdot q^k + F_{n,k-1} \cdot p^{n-k+1}$$

所以，我们有

$$F_{n,k} \cdot p^k + F_{n,k-1} \cdot q^{n-k+1} = F_{n,k} \cdot q^k + F_{n,k-1} \cdot p^{n-k+1}$$

也就是说，

$$F_{n,k} \cdot (p^k - q^k) = F_{n,k-1} \cdot (p^{n-k+1} - q^{n-k+1})$$

此外 $F_{n,0}=1$，所以如果 $p \neq q$，即 $p \neq 1/2$，我们就可以在 $\Theta(n)$ 的时间内算出所有的 $F_{n,k}$，从而得到答案。

而剩下的情况是 $p=q$，则此时和下标无关，任意两个人之间一个人胜利的概率都是 $1/2$。因此，我们考虑选出 k 个人作为最后的胜者，则有

$$F_{n,k} = \binom{n}{k} \cdot \frac{1}{2^{k(n-k)}}$$

两种情况的时间复杂度都是 $\Theta(n)$ 或 $\Theta(n \log_2 n)$。