

# 难题分享

samjia2000

2017 年 11 月 25 日

“这真的是两道不是很难的题目。”

给出一个长度为  $N$  的序列  $V$ ，问对于所有  $1 \leq i \leq j \leq N$ ，满足序列  $V$  的区间  $[i, j]$  中所有出现过的数字的出现次数都是奇数的  $(i, j)$  对数。

$$1 \leq N \leq 10^5$$

$$1 \leq V_i \leq 10^6$$

时间限制：3000ms

空间限制：128MB

设  $\text{ext}_{l,r}$  表示区间  $[l,r]$  中出现过的数字的状态,  $\text{pre}_k$  表示  $V_{1..k}$  中数字出现的次数的奇偶性的状态,  $\text{lst}_i$  表示满足  $j < i$  且  $V_j = V_i$  的最大的一个  $j$ 。

设  $\text{ext}_{l,r}$  表示区间  $[l, r]$  中出现过的数字的状态,  $\text{pre}_k$  表示  $V_{1..k}$  中数字出现的次数的奇偶性的状态,  $\text{lst}_i$  表示满足  $j < i$  且  $V_j = V_i$  的最大的一个  $j$ 。

对每种数字随机一个  $[0, 2^{64}]$  的权值, 然后上述的  $\text{ext}$  和  $\text{pre}$  就可以用 64 位整数表示了。

对序列分块。

枚举区间  $[i, j]$  的右区间，假设当前我们枚举到了  $i$ ，那么对于  $lst_i < l \leq i$  的  $ext$  都需要修改，查询个数的时候只要在每个块内查找等于  $pre_i$  的数的个数即可。

用 `hash` 处理，时间复杂度是  $O(n\sqrt{n})$

# An Unstable Graph

给出一个有  $n$  个点  $m$  条边的有向图，无重边，无自环，每秒第  $i$  条边出现的概率是  $p_i$ ，一开始  $A$  在  $x$  点，每一秒  $x$  要走一条存在的边，不可以不走，假如  $A$  绝顶聪明，问最后到达  $n$  的概率是多少。

$$n \leq 50$$

# An Unstable Graph

设  $f_i$  表示从点  $i$  到  $n$  的概率是多少。



# An Unstable Graph

设  $f_i$  表示从点  $i$  到  $n$  的概率是多少。对于一个点  $x$ ，计算  $f_x$  的策略应该是，按照他可以到达的点  $y$  的  $f_y$  从大到小排序，然后优先走  $f_y$  较大的。

# An Unstable Graph

设  $f_i$  表示从点  $i$  到  $n$  的概率是多少。对于一个点  $x$ ，计算  $f_x$  的策略应该是，按照他可以到达的点  $y$  的  $f_y$  从大到小排序，然后优先走  $f_y$  较大的。

证明很经典，只需要证明相邻两个  $y$  的大小关系即可。

# An Unstable Graph

设  $f_i$  表示从点  $i$  到  $n$  的概率是多少。对于一个点  $x$ ，计算  $f_x$  的策略应该是，按照他可以到达的点  $y$  的  $f_y$  从大到小排序，然后优先走  $f_y$  较大的。

证明很经典，只需要证明相邻两个  $y$  的大小关系即可。  
然后就可以用高斯消元解出所有的  $f_x$

# An Unstable Graph

设  $f_i$  表示从点  $i$  到  $n$  的概率是多少。对于一个点  $x$ ，计算  $f_x$  的策略应该是，按照他可以到达的点  $y$  的  $f_y$  从大到小排序，然后优先走  $f_y$  较大的。

证明很经典，只需要证明相邻两个  $y$  的大小关系即可。

然后就可以用高斯消元解出所有的  $f_x$

但现在的问题在于：我们不知道  $f_x$  的大小相对顺序，这导致我们无法直接做上面的过程。

# An Unstable Graph

我们可以在最开始先假定一个相对顺序。

然后根据这个相对顺序就可以用上面的过程算出  $f_x$ 。

但是这样做得到的  $f_x$  的相对顺序不一定就是上述的相对顺序。

如果恰好相同，那么就是找到了解。

如果不同，用算出的  $f_x$  的相对顺序来作为假定的相对顺序继续做上面的过程。

# An Unstable Graph

时间复杂度？

# An Unstable Graph

时间复杂度？

考虑每个  $f_x$  的初始排名和最终排名，由于每次得到的新的相对顺序都会向最后的排名逼近，所以每个点位置变化是  $O(n)$  的，故迭代的次数是  $O(n^2)$  的。

所以总的时间复杂度是  $O(n^5)$ 。