# RYOI2018R2题解

Cmd2001

# 写在前面

- 小X是谁？(话说这很重要吗)小X当然就是我啦(其实我并没有那么颓的说)。
- 感觉这次的题总体来说还是比较可做的。
- 部分分也给了很多档次，估计能拉开不小的差距呢。
- 不管是AK了还是爆零了，无论如何，开心就好。
- 毒瘤出题人什么的？ 不要闹，才不是啦！
- ~~话说标题的RYOI是什么意思？才不告诉你们！~~
- (其实就是指我家红酱了)

# T1

- 题意就是给你一个序列，问你区间内某个值出现次数，强制在线。
- 序列长度$\leq 7e5$，询问次数$\leq 7e5$。
- 时间限制$3s$，内存限制$40MB$。
- 具体数据范围：
- 对于20%的数据，$n,m \leq 5'000$，
- 对于40%的数据，$n,m \leq 50'000$，
- 对于70%的数据，$n,m \leq 500'000$，
- 对于100%的数据，$n,m \leq 7'00'000$。

# T1Sol

- 对于20%的数据，我会$O(n*m)$暴力！
- 对于40%的数据，我会树状数组套map。
- 好，想到树状数组套map的，这题可以GG了。因为你接下来会想写树套树，发现内存怎么也卡不过……
- 不过我们不妨把树套树内外层反转一下，反正颜色这一维度也不会查询区间是吧。
- 于是我们可以用map离散化颜色，对每个颜色开一棵动态开点线段树维护为这个颜色的点的出现位置。插入删除为$+1/-1$。
- 然而由于我卡了内存。这种做法仍然只有40分。

# T1Sol

- 考虑线段树肯定是不行了，我们能怎么办？写平衡树！
- 于是我们就得到了70分做法：对每个颜色开一棵平衡树，在对应位置插入删除。
- 时间复杂度$O(nlogn)$，空间复杂度$O(n)$。
- 正解是啥？
- 上面的就是正解了，但是需要卡常！
- 标程使用的是「非旋转treap」和「fread读入优化」。
- 注意这题平衡树一定要写GC，否则空间会被卡到$O(n+m)$级别，似乎也会MLE。

# 数据构造

- (这似乎是我的题解特有的部分，用来阐述我是怎么构造数据~~卡你们~~的)
- 对于全部的数据，预处理出一个长度为$n$，值域$\leq 1e9$的序列，作为备选颜色，之后的颜色直接下标访问。
- 对于20%的数据，颜色从[1,100]中随机生成。
- 对于另外20%的数据，颜色从[1,100]中随机生成。
- 然后有10%的数据，区间每个位置的初始颜色为它的下标，用来卡线段树的空间。
- 另外20%的数据，颜色从[1,5000]中随机生成。
- 最后30%的数据，颜色从[1,10000]中随机生成。

# T2

- 题意就是给你一个不明觉厉的期望DP……
- 同时包含修改，序列长度$\leq 1e5$，操作数量$\leq 1.5e5$。
- (这个真不能一句话题意了，一句话了就正解了)
- 具体数据范围见右表。
- 时间限制$1s$，空间限制$256MB$。

| 测试点编号 | n<= | m<= | 是否包含修改 |
|---|---|---|---|
| 1 | 3 | 300000 | N |
| 2 | 3 | 300000 | Y |
| 3 | 10 | 10 | N |
| 4 | 100 | 150000 | N |
| 5 | 100 | 150000 | N |
| 6 | 100 | 150000 | N |
| 7 | 100 | 10 | Y |
| 8 | 100 | 10 | Y |
| 9 | 100 | 10 | Y |
| 10 | 5000 | 150000 | N |
| 11 | 5000 | 150000 | N |
| 12 | 300 | 300 | Y |
| 13 | 300 | 300 | Y |
| 14 | 5000 | 5000 | Y |
| 15 | 5000 | 5000 | Y |
| 16 | 50000 | 50000 | N |
| 17 | 50000 | 50000 | N |
| 18 | 50000 | 50000 | Y |
| 19 | 50000 | 50000 | Y |
| 20 | 50000 | 50000 | Y |
| 21 | 100000 | 150000 | N |
| 22 | 100000 | 150000 | N |
| 23 | 100000 | 150000 | Y |
| 24 | 100000 | 150000 | Y |
| 25 | 100000 | 150000 | Y |

# T2Sol

- 显然期望DP最重要的就是状态转移方程。没有状态转移方程的话，什么都做不了啊。
- 我们定义$f_i$为从还剩下$i$个问题没有回答的状态，到我们的期望状态，所需要的步数。
- 那么我们有：$f_i = \frac{f_{i-1}}{c_i} + \frac{c_i - 1}{c_i} \frac{1}{n-i+1} \sum_{j=i}^{n} f_j + 1$
- 显然对于每次询问，我们期望从开始达到剩余$q$个问题的状态的话，我们有$f_q = 0$，而我们的答案，就是这个条件下的$f_n$。

# T2Sol

- 有了这个东西我们能干什么呢？我们能每次进行高斯消元，这样能通过6个测试点。
- 不过这样也太没有梦想了点……
- 我们不妨设$f_n = x$，显然我们可以用$kx + b$表示所有的$f_i$。
- 这样我们能一次高斯消元求出所有的所有的$f_i$，询问的时候直接带入$f_q = 0$求解$x$，只用在修改的时候全部重构。这样又能通过3个测试点了。

# T2Sol

- 等等，这么优美的公式为什么要高斯消元呢？
- 我们对转移方程进行变形：
- $f_i = \frac{f_{i-1}}{c_i} + \frac{c_i - 1}{c_i} \frac{1}{n-i+1} \sum_{j=i}^{n} f_j + 1$
- $c_i f_i = f_{i-1} + (c_i - 1) \frac{1}{n-i+1} \sum_{j=i}^{n} f_j + c_i$
- 假设我们已知了$f_{[i,n]}$，我们能否用这个公式推出$f_{i-1}$呢？
- $f_{i-1} = c_i f_i - (c_i - 1) \frac{1}{n-i+1} \sum_{j=i}^{n} f_j - c_i$
- 好的，我们又能通过4个测试点了。

# T2Sol

- 我们考虑后面的那个Σ是不是能优化成一个后缀和啊。
- $f_{i-1}=c_i f_i-(c_i-1)\frac{1}{n-i+1}\sum_{j=i}^{n} f_j-c_i$
- 我们替换下标
- $f_i=c_{i+1} f_{i+1}-(c_{i+1}-1)\frac{1}{n-i}\sum_{j=i+1}^{n} f_j-c_i$
- 我们定义：$s_i=\sum_{j=i}^{n} fj$
- 则我们有：
- $f_i=c_{i+1} f_{i+1}-(c_{i+1}-1)\frac{1}{n-i} s_{i+1}-c_{i+1}$
- 这样就可以$O(n)$转移了，又能通过6个测试点。

# T2Sol

- 观察这个转移，我们发现它是线性的，且$i$的状态只和$i+1$的状态有关。
- 我们能否把他写成矩阵呢？

```
fi+1.k          fi+1.b          si+1.k          si+1.b          1
*
ci+1            0               ci+1            0               0
0               ci+1            0               ci+1            0
-(ci+1-1)/n-i   0               -(ci+1-1)/n-i+1 0               0
0               -(ci+1-1)/n-i   0               -(ci+1-1)/n-i+1 0
0               -ci+1           0               -ci+1           1
=
fi.k            fi.b            si.k            si.b            1
```

你们就自行脑补我写的这是矩阵就好了，注意所有的$c$都是$c_{i+1}$，此外所有的$n-i$都为最先计算。

# T2Sol

- 如果你用分块维护，可以多通过3个测试点。
- 如果你用线段树维护，就可以AC啦！

# 数据构造

- 这题数据直接随机就好了吧。
- 对于没有修改的测试点，我们钦定操作是询问即可。

# T3

- 无非就是给你一张图，让你求最小边覆盖(用最少的边覆盖所有的点)。
- 数据范围点数$\leq 500$，边数$\leq 1000$。
- 时间限制$1s$，空间限制$128MB$。
- 具体数据范围：
- 对于前5%的数据，$n \leq 5$。
- 对于另外15%的数据，$n \leq 10$。
- 对于另外20%的数据，$n \leq 16$。
- 对于另外30%的数据，存在边的两个点奇偶性不同。
- 对于100%的数据，$n \leq 500, m \leq 1000$。

# T3Sol

- 显然答案就是n-最大匹配。
- 对于5%的数据，我会手玩~~(其实输出样例就可获5分)~~！
- 对于20%的数据，我会枚举方案爆搜！
- 对于40%的数据，我会状压子集DP，或者枚举每个点在左边或是右边然后求最大匹配！
- 对于另外20%的数据，我会二分图最大匹配匈牙利！
- 正解？难道真的写一般图最大匹配的带花树吗？

# T3Sol

- 然而并不用！
- 一般图最大匹配是存在一个随机化算法的。
- 就是在二分图最大匹配算法的基础上，每次随机排序出边，然后让互相匹配的两个点存储的匹配状态同步。
- 它是基于这样一个事实：如果我们有三个点1,2,3，原状态为1,2 – 3，那么我们把状态改为1 – 2,3至少不会让答案更劣。
- 即使图中这三个点在一个奇环上。
- (然而这个算法还是有出错概率的，解决方案就是多跑几遍，反正在UOJ范围内能获得AC)

# 数据构造

- 除了前5分的测试点和二分图的测试点。
- 其余的测试点都是我写了一个二分图匹配匈牙利和正解对拍拍出来的。话说直接随机数据的话，写个匈牙利就差不多能AC啦。
- $zhy$大佬有一个能通过前40分的骗分网络流，你们可以向他请教一下。

# 关于题目的idea

- T1的idea主要源于51nod上某题。
- 那题正解空间$O(n)$，而我一开始却想写替罪羊套map然后被卡空间……
- 于是我把那题弱化了一下，就有了这道题
- T2的idea源于~~《君彼女》~~的无限循环……
- 不需要解释，懂的自然懂。
- 只不过那个东西也没有这题意这么鬼畜(其实这个题是能做到询问任意状态到任意状态的，只不过出给你们就人干事了)。
- T3的idea源于4月份我在北京集训的时候学到的一般图最大匹配的科技(当时还不知道这是早就有的，我们都以为这是那个同学发明的，简直火星救援)。

# 题目总结

- 相当中规中矩的三道题吧。
- 曾有人说过：三流选手凭知识，二流选手比姿势，一流选手拼意识。
- T1大概就是一道意识题，考察大家计算内存占用的意识和必要的常数优化的意识(相信一定有不算空间MLE的)。
- T2的话，感觉是姿势题，考察大家思维能力(智商检测题)，并没有多少码量。
- T3？大概也算是意识题吧，考察大家分SubTask写暴力或者乱搞骗分的意识~~(其实我本意是让分SubTask暴力的和骗分的拿到70，直接无脑二分图的WA一堆点的)~~。~~正解的随机化似乎也不难想到吧~~。
- 最后，祝大家玩得开心！

# 国际惯例的

*谢谢大家*