Fortune Telling 2 解题报告

绍兴市第一中学 任之洲

1 试题来源

JOI Open Contest 2013/2014 Day 1

2 试题大意

有N张牌，每张牌有两面，分别写着A_i和B_i，一开始所有牌A_i朝上放置。

有K个操作，每个操作有一个参数T_j，这次操作会将所有朝上面点数不超过T_j的牌都翻面。

求经过所有K次操作后，所有牌朝上面的点数之和。

2.1 数据规模与约定

$1 \le N, K \le 200000$

$1 \le A_i, B_i, T_j \le 10^9$

- subtask1 [4points]: $1 \le N, K \le 1000$
- subtask2 [31points]: $1 \le N, K \le 40000$
- subtask3 [65points]: 没有额外的限制

时间限制2s，空间限制256MB

3 算法介绍

3.1 算法一

根据题意可以采取两种朴素实现手段：

- 对于每个操作，枚举每张牌判断是否需要翻转。
- 对于每张牌，依次枚举每个操作判断是否需要翻转。

这两种实现的时间复杂度均为$O(NK)$，可以通过subtask1。

由于牌的顺序并没有实际的意义，所以选择采取第二种实现来优化。

3.2 算法二

考虑将每S个操作分为一块，计算每张牌经过这S个操作后是否翻转。

一张牌是否翻转只和牌的两面点数A_i, B_i与这S个操作参数T_j的相对大小关系有关，权值将会被这S张牌划分为$O(S)$段区间，所以本质不同的牌总共有$O(S^2)$种。

对于每个块，可以枚举这$O(S^2)$种情况，预处理出每一种牌经过这个块后的翻转情况，每个块的时间复杂度为$O(S^3)$，块的个数为$O(\frac{K}{S})$，所以这一部分的复杂度为$O(KS^2)$。

对于每张牌，需要依次经过这$O(\frac{K}{S})$个块，得到它的翻转情况。为了确定这张牌的两面点数属于这个块的哪一类，就需要得到A_i, B_i与块中T_j的相对关系。

如果选用二分来确定，那么每一块的复杂度为$O(N\log S)$。

可以预先将A_i和B_i分别排序，那么每个块就可以利用递增关系来做到$O(N+S)$确定相对关系，这一部分的复杂度为$O(N\log N+\frac{NK}{S}+K)$。

整理一下，总时间复杂度即为$O(N\log N+\frac{NK}{S}+KS^2)$，所以$\frac{NK}{S}=KS^2$即$S=N^{\frac{1}{3}}$时较优。

时间复杂度$O(N\log N+N^{\frac{2}{3}}K)$，空间复杂度$O(N+K)$，可以通过subtask2。

3.3 算法三

对于一张牌A_i, B_i，对它来说本质不同的操作只有三种：

$$T_a \ < \ min(A_i, B_i) \ \le \ T_b \ < \ max(A_i, B_i) \ \le \ T_c$$

这三种操作对这张牌的影响如下：

- T_a：两面都大于T_a，可以直接忽略这个操作。
- T_b：只有当前面是较小面的时候需要被翻面。

- T_c：两面都不超过T_c，不管现在的状态怎样都一定会翻面。

容易发现，经过一个T_b类操作后，这张牌一定是较大面朝上。

不妨对每张牌求出最后一个T_b类操作，经过这个操作后，这张牌一定翻到了较大面，在那之后只要统计一下T_c类操作的数量就能确定最终状态。

那么，剩下两个子问题：

- 找到最后一个$T_j \in [min(A_i, B_i), max(A_i, B_i))$的操作，设为$last$。

- 统计有多少$j \in (last, K]$满足$T_j \geq max(A_i, B_i)$。

将T_j离散化后，第一个子问题就等同于对一段权值区间求最大值，可以用线段树维护。第二个子问题是一个矩形二维数点问题，可以将询问按照$last$排序扫描，用线段树维护。

时间复杂度$O((N + K) \log K)$，空间复杂度$O(N + K)$。