FrozenStandings 解题报告

浙江省镇海中学 杜瑜皓

1 试题来源

Topcoder Open 2014 Finals 1100pts by rng_58

2 试题大意

一场比赛有n道题，最后一个小时封板，封板前每个人通过了x_i个题，封板后每个人至多通过一个题，所以最后可能通过的题数为x_i或x_i+1。

问最后有多少种不同的排名。如果题数相同按编号排序。

规定$n \le 5*10^5$。

3 算法介绍

首先它是个双关键字排序，可以将每个人的成绩加权为$-x_i*w+i$，w是一个很大的整数，那么双关键字排序变成一个关键字。每个人对应的分数即为r_i，那么它最后可能得到的分数为r_i或者r_i-w，记$l_i=r_i-w$。

为了方便，可以将l_i排序，也就是保证$l_1<l_2<\cdots<l_n$。

接着考虑如何求不同的排名。每个人选r_i或者l_i，那么有两种选择，总共有2^n种选择。那么要去掉其中重复的排名。

如果有两种选择方案，存在一个人选择l_i和r_i，并且l_i和r_i之间没有一个数字被选择，那么这两种方案是一样的，在总方案中要减掉其中一种。

如果一个人对应的l_i,r_i这段中没有一个数字被选择，那么对于j满足$l_i<l_j<r_i$或$l_i<r_j<r_i$，必须选择左边的或者右边的。

因为区间的长度全都相同，那么这样的j也构成了一个区间，即为$[L_i,R_i]$。

令dp_k表示$1\sim k$这些人任意选择能构成的不同的排名，考虑最后一个人选择左或右，那么有两种取值方案，同时我们要减去前面所说的矛盾的情况。

对于每个人i，如果它产生矛盾，那么$[L_i, R_i]$这一段内的方案都是确定的，那么只要在dp_{R_i}处减去dp_{L_i-1}即可。

所以$dp_k = 2dp_{k-1} - \sum_{R_i=k} dp_{L_i-1}$。

只要使用two pointer来实现，就可以做到$O(n)$。

4 总结

这是一个很难的题，使用容斥原理减掉重复的排名。但是这里没有明显的限制和切入口，需要去发现。同时处理手法很奇怪，对于一个人只在他影响的区间最后去掉他的贡献。当然它的做法相当简洁。

我做这个题使用了一个相当复杂的做法，我的想法是考虑一些极长的子段可以左右移动，如果子段长度为l，那么要去掉l中不同的方法。然后这些子段两端都有一些奇怪的条件限制，由于过于复杂，不再详细描述。