CodeChef September Challenge 2014 解题报告

大连市第二十四中学 于纪平

目 录

0 概要

这一场CodeChef Long Challenge的有些题目非常有趣，在此来分享一下详细的中文解题报告。

比赛页面：http://www.codechef.com/SEPT14。

1 UASEQ: Chef and sequence

1.1 题目大意

完整题面：http://www.codechef.com/SEPT14/problems/UASEQ。

给出一个n个元素的整数序列，要求修改至多k个元素（修改后的元素必须还是整数），使得序列变为等差数列，要求首项最小的前提下，公差最小。

$n \le 10^5, k \le 10, k \le n-2$。保证存在方案。

1.2 算法

每次随机选取两个位置不修改，修改其他位置使得序列成为等差序列，如果修改的元素小于k个就尝试更新答案。

考虑这个算法的正确率：对于n较小的情况，由于最优解至少有2个位置不修改，我们几乎必然会随机到这个位置；对于n较大的情况，由于最优解至少有$n-k$个位置不修改，我们几乎必然会随机到这$n-k$个位置的任意一对。

无论如何正确率都是可以保证的，所以这道题就这么水过了。

2 SEABUB: Sereja and Bubble Sort

2.1 题目大意

完整题面：http://www.codechef.com/SEPT14/problems/SEABUB。

给出一个长度为n的排列，每次可以交换两个相邻的数或将排列随机打乱，问在不超过k次操作的情况下最优策略最终序列的逆序对的期望（最小值为最优）。

$n \le 100, k \le 10^{18}$，数据组数不超过100。

2.2 算法

最优策略应该是这样的：如果认为当前的逆序对数在随机排列中比较小则进行k次冒泡操作；否则进行一次随机打乱操作。

设$best[i]$表示，如果还剩i次交换，应该让逆序对数不超过$best[i]$的序列做冒泡操作。

设$avg[i]$表示，如果当前序列是随机序列，并且还有i次交换机会，那么最优策略下最终序列的逆序对的期望。那么根据定义，$best[i]$是满足$best[i] - i \le avg[i-1]$的最大整数，所以$best[i] = \lfloor avg[i-1] + i \rfloor$。

考虑$avg[i]$的求法。设当前序列逆序对数为j，那么如果$j \le best[i]$，答案应该是$\max(j-i, 0)$，否则答案应该是$avg[i-1]$。

设$P[i][j]$表示，长度为i的随机序列，逆序对数为j的概率。那么

$$avg[i] = \sum_{j \le best[i]} P[n][j] * \max(j-i, 0) + \sum_{j > best[i]} P[n][j] * avg[i-1]$$

先考虑P的求法。显然$P[i][j] \sum k = 0^{i-1} P[i-1][j-k]$，用简单的前缀和优化就可以做到$\Theta(n^3)$预处理。

直接按照上面的式子计算avg是每次询问$\Theta(n^4)$的时间复杂度。我们将avg的式子改写为：

$$avg[i] = \sum_{i < j \le best[i]} j * P[n][j] - i \sum_{i < j \le best[i]} P[j] + avg[k-1] \sum_{j > best[i]} P[j]$$

利用这个式子，只再处理一个$j * P[n][j]$的前缀和就可以快速计算avg了。

计算出P、avg和$best$以后，只需比较询问序列的逆序对数是否小于等于$best[k]$。如果是则输出逆序对数减k，否则输出$avg[k-1]$。

3 QRECT: Rectangle Query

3.1 题目大意

完整题面：http://www.codechef.com/SEPT14/problems/QRECT。

q次操作，每次可以在坐标平面上插入一个与坐标轴平行的矩形，或删除一个矩形，或询问一个矩形与多少个矩形有公共点。

$q \le 10^5$。

3.2 算法

首先把坐标范围离散化成$O(q)$的。

令插入矩形的权值为1，那么删除操作等价于加入一个权值为-1的矩形。

对于每个询问考虑它的反面，即询问这个矩形与多少个矩形没有公共点。

考虑已知矩形A和询问矩形B，如果它们没有公共点，说明A完全在B的某个方向（上下左右）。那么对于矩形B，我们对于四个方向分别统计完全在B的这个方向的已知矩形的权值和。这样会把四个角上的已知矩形统计两遍，所以还要依次减去。

对于每个询问矩形B我们要查询8个值，注意到这个显然是旋转对称的，我们只考虑一个方向，即维护：完全在B左边的矩形个数，减去完全在B左下方的矩形A的权值之和。

不难得出，这个条件等价于，A的右上角在B的左下角的左方或左上方。也就是说对于这个问题，每个矩形我们只需要考虑一个点。

问题转化为了，在坐标平面上支持修改点权或求一个与坐标轴平行的矩形的点权之和。这是经典问题，可以用按时间分治后扫描线，或者树套树等经典方法处理。

4 FIBTREE: Fibonacci Numbers on Tree

4.1 题目大意

完整题面：http://www.codechef.com/SEPT14/problems/FIBTREE。

给出一棵树，树上有点权（初始为0）。定义Fibonacci数列$F_1 = F_2 = 1, F_n = F_{n-1} + F_{n-2}$。

有m次操作，每次可以：

- A x y表示，找出从x到y的树上路径，对路径上的第i个点的点权增加F_i。
- QS x y表示，求如果以x为根建立有根树，则y子树的点权之和。
- QC x y表示，问从x到y的树上路径的点权之和。
- R x表示，所有点权回到第x操作后的状态。

所有询问答案对$10^9 + 9$取模输出，$n, m \le 10^5$，强制在线。

4.2 简化版

这道题乍一看非常麻烦，我们先考虑它的简化版，即链上的情况。要求支持区间覆盖Fibonacci数列，以及求区间元素的和。

首先想到线段树，然后考虑打标记。

我们定义广义Fibonacci数列：前两项为任意数，从第三项开始满足$F_n = F_{n-1} + F_{n-2}$。显然，广义Fibonacci数列可以通过其前两项唯一确定；两个广义Fibonacci数列的和还是广义Fibonacci数列。

我们在区间上打的标记要记录这个区间被哪个广义Fibonacci数列覆盖，则只需记录其前两项。

两个标记的合并非常简单，直接加起来就可以了。

标记下传方法是：向左孩子传标记无须改动，向右孩子传标记需要将某个广义Fibonacci数列递推若干项。这只需处理广义Fibonacci数列的转移矩阵的0到n次幂就可以了。

标记对统计量的影响也是很好计算的。我们需要计算一个广义Fibonacci数列的前若干项和，只需要在转移矩阵里面再算一个前缀和就可以了。

这样我们就解决了链上的操作。如果需要支持回退操作，只要把这个线段树进行可持久化就可以了。

4.3 算法

对于树上的问题，只需对树进行轻重链剖分，套用之前的方法就可以解决A和QC，进行可持久化就可以解决R。

考虑如何做QS。显然这里并不需要真的换根。假设以1为根，那么对于QS x y，如果y不是x的祖先，那么结果就和QS 1 y是一样的；如果y是x的祖先，那么设t是y的某个儿子且是x的祖先，那么结果就是(QS 1 1)-(QS 1 t)。特殊地，如果x就是y，那么结果就是QS 1 1。

总之我们实际上只需要维护以1为根的情况下每个子树的点权和。

我们直接维护答案。设$sum[i]$表示i子树的点权和。考虑一次A操作对sum的影响：一次A操作可以根据LCA拆成2次从某个点到它到根路径上某个点（或反之）的A操作。每个这样的操作都只会影响一条链上的sum值。

直接维护sum，问题就转化为了：每次可以给一条链的sum覆盖上一个广义Fibonacci数列的前缀和，或查询一个点的sum。这可以轻重链剖分用线段树维

护，方法与上面类似，只是维护的矩阵不同。而且因为是单点查询，所以可以标记永久化，不用下传，减小了空间常数。

这道题现在就应该可以做了，但是维护QC的答案依然需要标记下传。我们考虑每一棵维护QC的的线段树，这个线段树需要支持：区间覆盖一个广义Fibonacci数列，区间求和。我们只需维护这个序列的前缀和，就把问题变成了：区间覆盖一个广义Fibonacci数列的前缀和，单点查询。这样问题就与维护QS一样了，也可以标记永久化。

注意：维护QS是先在树上做（从叶到根的）前缀和，再树链剖分；维护QC是先树链剖分，然后在剖出来的每条链上做前缀和。如果查询QS，只需要在一棵线段树上查询；如果要查询QC，还要按树链剖分的做法一步步走到根上去。

总之，最后的时空复杂度均为$O(q\log^2 n)$，可以通过本题。