Bestcoder Round #19 解题报告

大连市第二十四中学 于纪平

目 录

0 概要

这一场BestCoder的题目是我出的，在此来分享一下详细的中文解题报告。比赛页面：http://bestcoder.hdu.edu.cn/contests/contest_show.php?cid=551。

1 1001 Alexandra and Prime Numbers

1.1 题目大意

完整题面：http://acm.hdu.edu.cn/showproblem.php?pid=5108。

给出正整数n，求最小正整数m使得n/m是质数，或输出无解。

$n \le 10^9$，至多有100组数据。

1.2 算法

要使m最小，n/m应该是n的最大质因子。暴力找出最大质因子后，把它除n就是m了。

唯一无解的情况是$n = 1$。

2 1002 Alexandra and A*B Problem

2.1 题目大意

完整题面：http://acm.hdu.edu.cn/showproblem.php?pid=5109。

给出正整数a和数字串s，求最小正整数b，使得s是$a \times b$的十进制表示串的子串。

$a \le 10000, |s| \le 8$，至多有100组数据。

2.2 算法

$a \times b$的结果应该是xsy的形式，其中串y可以为空，在s不以0开头的情况下x也可以为空。

首先枚举y的长度。假设$|y|$已经确定，那么x显然越小越好，所以再从小到大枚举x。可以得到方程：

$$10^{|s|+|y|}x + 10^{|y|}s + y \equiv 0 \pmod a$$

这是关于y的一元一次模方程，我们应该取最小的y，并且如果$y < 10^{|y|}$才符合题意。

这个方程关于x也是一次的，所以x只需枚举到a，如果依然没有解就永远不会有解了。

然后考虑$|y|$的枚举范围。注意$|y| = 0$时显然能够找到一个$x < 10000$的解，所以$|y|$只需枚举到4就可以了。

对于枚举出的4个答案取最优解就可以了。时间复杂度$O(a \log a)$。

3 1003 Alexandra and COS

3.1 题目大意

完整题面：http://acm.hdu.edu.cn/showproblem.php?pid=5110。

给出一个$n \times m$的01矩阵A，q次询问，每次给出x, y, d，求

$$\sum_{i=1}^{n}\sum_{j=1}^{m} [x - i \geq 0]\,[d|(x-i)]\,[y-(x-i) \leq j \leq y+(x-i)]\,A_{i,j}$$

$n, m \leq 1000, q \leq 500000$。

3.2 算法

对每一行处理前缀和，即处理$S_{i,j} = \sum_{k=1}^{j} A_{i,k}$。

考虑询问的式子，这个式子实质上是从(x, y)出发，每次向上移d个单位，同时让询问的左右端点向两端各扩展d个单位后询问一行的部分和。部分和可以$O(1)$计算，所以直接暴力每次的时间复杂度是$O(n/d)$。

暴力算法在d较大时是很快的。对于d较小的情况，考虑直接预处理出所有的答案。设答案是$ans(x, y, d)$，那么它可以用$ans(x-d, y-d, d)$和前缀和数组迅速算出。

设我们预处理出$d \leq k$时的所有答案，那么预处理的时间复杂度是$O(n^2k)$，所有询问的时间复杂度是$O(qn/k)$。

为了使总时间复杂度最小，取$k = \sqrt{q/n}$，总时间复杂度为$O(n^{1.5}q^{0.5})$。

4 1004 Alexandra and Two Trees

4.1 题目大意

完整题面：http://acm.hdu.edu.cn/showproblem.php?pid=5111。

给出两棵树，有点权，保证同一棵树中任意两点的点权不同。现在有q次询问，每次给出u_1, v_1, u_2, v_2，问第一棵树上u_1到v_1的树上路径与第二棵树上u_2与v_2的树上路径有多少个相同的数。

$n \le 100000, q \le 50000$。

4.2 算法1

根据经典的把树上路径问题转化为某个点到根的路径问题的方法，设询问的答案是$ans(u_1, v_1, u_2, v_2)$，设1为根，u_i和v_i的LCA是l_i，那么有

$$ans(u_1, v_1, u_2, v_2) = ans(1, u_1, u_2, v_2) + ans(1, v_1, u_2, v_2) - 2ans(1, l_1, u_2, v_2)$$

同理

$$ans(u_1, v_1, u_2, v_2) = ans(u_1, v_1, 1, u_2) + ans(u_1, v_1, 1, v_2) - 2ans(u_1, v_1, 1, l_2)$$

不难得出

$$\begin{aligned} ans(u_1, v_1, u_2, v_2) = & ans(1, u_1, 1, u_2) + ans(1, u_1, 1, v_2) - 2ans(1, u_1, 1, l_2) \\ & +ans(1, v_1, 1, u_2) + ans(1, v_1, 1, v_2) - 2ans(1, v_1, 1, l_2) \\ & -2ans(1, l_1, 1, u_2) - 2ans(1, l_1, 1, v_2) + 4ans(1, l_1, 1, l_2) \end{aligned}$$

这样我们就把一个询问拆成了9个询问，每次给出u_1, u_2，问第一棵树上1到u_1的路径与第二棵树上1到u_2的路径上相同数的个数。

考虑在第一棵树的每个节点都开一棵线段树，节点u_1上的线段树的定义域是第二棵树的节点集合，这棵线段树下标为u_2的位置存的是询问(u_1, u_2)的答案。

在第一棵树上形成父子关系的两个节点上的线段树会有若干个不同的元素，但是由于第二棵树上至多只有一个节点与子节点权值相同，所以子节点的线段树只有某个第二棵树的子树位置与父亲线段树相比多了1，这可以通过对第二棵树取dfs序后，将子树映射到一个连续区间，通过一次修改完成。（如果第二棵树上没有节点的点权与孩子相同，则两棵线段树完全相同）我们可以用可持久化线段树将这些线段树全部存下，再按上式查询即可。

同理，也可以离线处理，按照u_1在第一棵树上的dfs序排序，依次处理每个询问，与上面的方法本质是一样的。

时间复杂度为$O((n+q)\log n)$，但常数较大（9个询问）。

4.3 算法2

先考虑链上的问题：有两个序列，每次在两个序列中各取一个区间，问两个区间相同的数的个数。

由于每个序列都不会有相同的数，我们显然能找到一种方法把第一个序列变成$[1,2,\cdots,n]$，第二个序列也做相应变化使得所有询问的答案不改变。

问题转化为了，每次给出l_1,r_1,l_2,r_2，问第二个序列的$[l_2,r_2]$区间中有多少个数落在$[l_1,r_1]$中。这是数据结构经典题目，可以用线段树离线求，或可持久化线段树在线求。

对于树上的问题，我们几乎不可能对第一棵树进行若干变换使得所有询问的(u_1,v_1)路径的权值都是某个连续区间，但是我们可以对第一棵树根据轻重链剖分重新标号，使得所有询问的(u_1,v_1)的权值都可以写成$O(\log n)$个不交区间的并。

问题转化为了，每次给出l,r,u,v，问第二棵树的(u,v)路径上有多少个点权落在$[l,r]$中。这与序列上的问题类似，可以用线段树离线后按dfs序求，或树上递推可持久化线段树在线求。

一共有$O(q\log n)$个转化后的询问，所以总时间复杂度为$O(q\log^2 n)$，但常数较小（轻重链剖分）。