某 AFO 狗の NOI 模拟题题解

by jiry_2

June 28, 2016

竞赛时长：5 小时

题目名称	数	树	数据结构
输入文件名	number.in	tree.in	data.in
输出文件名	number.out	tree.out	data.out
每个测试点时限	1s	2.5s	2.5s
测试点数目	10	10	10
每个测试点分值	10	10	10
内存限制	512MB	512MB	512MB
是否有部分分	否	否	否
题目类型	传统	传统	传统
是否有 SPJ	否	否	否

1 数

因为 10^K 一定是 2^K 的倍数，所以给一个正数加上 10^K 后，它的二进制表示的后 K 位一定是不变的。

令 S_k 为所有满足 $x < 10^k$ 且 x 的十进制表示的后 k 位于二进制表示的后 k 位相同（不足补 0）的数字 x 的集合。我们可以简单地从 S_k 扩展出 S_{k+1}，只要枚举第 $k+1$ 位是 0 还是 1 然后判断二进制表示的第 $k+1$ 位是否相同即可。

要支持的操作有给一个数加上 10^k，询问一个数二进制表示下的第 K 为值。可以在 2^a 进制下用高精度存储 S_k 中的数，预处理出 10^k 在这个精度下的表示，这样就能做到第一个操作 $O(\frac{L}{a})$，第二个操作 $O(1)$，L 是数字长度。

这样的时间复杂度大约是 $O(\frac{nL}{a})$ 的，标程取了 $a = 25$。

2 树

因为树是二分图，所以树的最大独立集大小等于节点数减去最大匹配。

求树的最大匹配是一个经典的 DP 问题，令 $f[i]$ 为第 i 个点的子树中选了第 i 个点的最大匹配数，$g[i]$ 为不选第 i 个点的最大匹配数。

因为这题要求计数，所以可以记 DP 值为状态（就是 DP 套 DP）。直接压的话状态数是 $O(n^3)$ 的，有点难以接受。不过不难发现在求最大匹配的 DP 过程中，DP 值一定满足 $0 \le f[i] - g[i] \le 1$，所以可以令 $F[i][j]$ 为 $i+1$ 个点，根节点的 g 值为 j，f 值为 $j+1$ 的树的个数，$G[i][j]$ 为 $i+1$ 个点，根节点的 g 值为 j，f 值为 j 的树的个数。考虑每一次给根节点接上一个新的子树，就能得到转移方程，具体如下：

$$G[i][j] = \sum_{a=1}^{i} \sum_{b=0}^{j} G[i-a][j-b] \times F[a-1][b-1] + [i=0, j=0]$$
$$F[i][j] = \sum_{a=1}^{i} \sum_{b=0}^{j} F[i-a][j-b] \times (G[a-1][b] + F[a-1][b-1]) + G[i-a][j-b] \times G[a-1][b]$$

直接动态规划的时间复杂度是 $O(n^4)$。我们可以用 FFT 来优化第二重循环，这样时间复杂度就变成了 $O(n^3 \log n)$。

更进一步地，在 DP 时我们可以直接使用 DFT 后的点值进行运算，在最后用 n 次 IDFT 还原出答案数组，时间复杂度 $O(n^3 + n^2 \log n)$，可以通过所有测试点。

实际上存在复杂度更低的算法，但是因为常数比较大，所以难以在合理的时限范围内与 $O(n^3)$ 算法产生区分（好像一个点开一分钟的时限还是能卡掉的）。

可以将 G 和 F 看成关于节点数 x，根节点的 g 值 y 的形式幂级数，可以得到方程：

$$G = xyFG + 1$$
$$F = xFG + xyF^2 + xG^2$$

二元形式幂级数比较难处理，但是可以发现 y 的值永远不会超过 $\lfloor \frac{n}{2} \rfloor$，所以可以令 $x = y^{\lfloor \frac{n}{2} \rfloor + 1}$，这样就变成了一元的问题。

有了这个方程后可以直接使用分治 FFT 的方法求出两个形式幂级数的前若干项系数。时间复杂度 $O(n^2\log^2 n)$。

当然也可以选择解这个方程，带入后可以得到：

$$\frac{xG(G-1)}{xyG}+xy(\frac{G-1}{xyG})^2+xG^2-\frac{G-1}{xyG}=0$$

转化成一维问题后牛顿迭代一波就能得到答案了，时间复杂度 $O(n^2\log n)$。

3 数据结构

考虑如何计算一个序列的权值。我们让每一个子序列都在其第一次出现的位置（下标字典序最小的出现位置）被统计，这样就不用去重了。令 $dp[i]$ 为在第 i 位被统计的子序列个数，令 $ne[i][j]$ 为第 i 个位置后第一次出现数字 j 的下标，那么 $dp[i]$ 可以更新到所有 $dp[ne[i][j]]$。这个字符串的权值为 $\sum_{i=1}^n dp[i]$。

这样暴力 DP 单组询问的时间复杂度是 $O(nK)$，考虑用数据结构优化这个这个 DP。对序列建线段树，每一个节点 i 记录 $f[i][u][v]$ 表示这个区间中开头为 v 的子序列，更新到区间后第一个 u 的和。那么合并两个区间就相当于进行一次矩阵乘法。

在给区间加上一个数时，区间中任意两个数的相等关系并不会变，因此只需要给矩阵的行列置换一下就好了，内容本质上并没有变化。

在给区间乘上一个数 x 时，如果 $(x,K)=1$，相等关系同样不会变化，处理方法和区间加一样。对于其他情况，区间中的数以模 $\frac{K}{(x,K)}$ 的余数划分成了 $\frac{K}{(x,K)}$ 个等价类，这样就不能直接用原有的值更新了。处理方法是我们对 K 的所有约数 d，都用一棵线段树维护模 d 时的答案，在区间乘法时相当于用模 $\frac{d}{(x,d)}$ 的信息去更新模 d 的信息。

时间复杂度 $O(nK^3\log n)$