晚间划水

HwannaKgun

2017 年 3 月 7 日

扯淡

- 难度不够，数量来凑
- 相信很多同学都能在几分钟内有完整思路
- 让我们荡起双桨

- 给出一个长度为 n 的序列 $\{S\}$，现将这个序列的所有不同排列按字典序排序，求 $\{S\}$ 的排名模 m 的值
- $n \leq 3 \times 10^5$
- 洋栗，给出序列 $\{2, 1, 10, 2\}$，$m = 1000$；答案为 5

- 不保证 m 为质数，所以先将 m 分解，对每个 $p_i^{q_i}$ 单独计算答案后 CRT 合并之
- 然后是套路时间
- 从左往右依次考虑每一位对答案的贡献
- 下面从第一位开始考虑

- 设一共有 k 种数，从小到大分别为 $a_1, a_2, \ldots, a_k$，各有 $t_1, t_2, \ldots, t_k$ 个
- 设 p 是使 $a_p < S_1$ 成立的极大的 p
- 只要待填排列的第一位从这前 p 种数里选，剩下的 $n-1$ 个数就可以随便排列
- 第一位对答案的贡献就是

$$\frac{(n-1)!}{\prod_{i=1}^{k} t_i!} \cdot \sum_{j=1}^{p} t_j$$

- 于是 $\sum_{j=1}^{p} t_j$ 可以快乐地前缀和

- 设一共有 k 种数，从小到大分别为 $a_1, a_2, \dots, a_k$，各有 $t_1, t_2, \dots, t_k$ 个
- 设 p 是使 $a_p < S_1$ 成立的极大的 p
- 只要待填排列的第一位从这前 p 种数里选，剩下的 $n-1$ 个数就可以随便排列
- 第一位对答案的贡献就是

$$\frac{(n-1)!}{\prod_{i=1}^{k} t_i!} \cdot \sum_{j=1}^{p} t_j$$

- 于是 $\sum_{j=1}^{p} t_j$ 可以快乐地前缀和

- 设一共有 k 种数，从小到大分别为 $a_1, a_2, ..., a_k$，各有 $t_1, t_2, ..., t_k$ 个
- 设 p 是使 $a_p < S_1$ 成立的极大的 p
- 只要待填排列的第一位从这前 p 种数里选，剩下的 $n-1$ 个数就可以随便排列
- 第一位对答案的贡献就是

$$\frac{(n-1)!}{\prod_{i=1}^{k} t_i!} \cdot \sum_{j=1}^{p} t_j$$

- 于是 $\sum_{j=1}^{p} t_j$ 可以快乐地前缀和

- 第一位对答案的贡献为 $\frac{(n-1)!}{\prod_{i=1}^{k} t_i!} \cdot \sum_{j=1}^{p} t_j$
- 接下来的思路就更显然了
- 从小到大考虑 $i\,(i > 1)$：对于前 $i-1$ 位与 S 相等，第 i 位比 S_i 小的情形，我们需要高效地获得当前可用的每一种数字的数量
- 相当于前 $i-1$ 位已被钦定，直接改 t 数组即可
- 所以我们要做的就是 t_i 的单点修改和求前缀和
- 树状数组维护

- 第一位对答案的贡献为 $\frac{(n-1)!}{\prod_{i=1}^{k} t_i!} \cdot \sum_{j=1}^{p} t_j$
- 接下来的思路就更显然了
- 从小到大考虑 $i\ (i > 1)$：对于前 $i-1$ 位与 S 相等，第 i 位比 S_i 小的情形，我们需要高效地获得当前可用的每一种数字的数量
- 相当于前 $i-1$ 位已被钦定，直接改 t 数组即可
- 所以我们要做的就是 t_i 的单点修改和求前缀和
- 树状数组维护

- 第一位对答案的贡献为 $\frac{(n-1)!}{\prod_{i=1}^{k} t_i!} \cdot \sum_{j=1}^{p} t_j$
- 接下来的思路就更显然了
- 从小到大考虑 $i\ (i > 1)$：对于前 $i-1$ 位与 S 相等，第 i 位比 S_i 小的情形，我们需要高效地获得当前可用的每一种数字的数量
- 相当于前 $i-1$ 位已被钦定，直接改 t 数组即可
- 所以我们要做的就是 t_i 的单点修改和求前缀和
- 树状数组维护

- CRT 合并? 模 $p_i^{q_i}$ 怎么算?
- 一个常用黑科技 (?)，把分子分母都表示成 (a, b) 整数二元组的形式，意义是 $x = (a, b) = a \cdot p^b$，其中 b 极大
- 这样就保证了 a 的逆元一定存在，可以开心地 exgcd 咯
- 总的时间复杂度 $O(?n \log n)$，其中 ? 是 m 的质因子数目

Binary vs. Decimal
source: NEERC 2015

- #define 稳数：一种十进制正整数，满足它本身是其二进制表示的后缀
- 求第 n 个稳数，$n \leq 10^4$
- 洋栗，前 10 个稳数分别是 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1100
- 可以用一些奇技淫巧打表，但中央钦定不允许这么干

- 好想成为暴力大师
- 朴素的 TLE 想法是 从低位往高位 BFS 枚举加 0/1 判断
- 但暴力大师看样例打表发现性质：一个稳数的任意后缀一定还是稳数
- 反过来，如果当前数不是稳数，就可以减掉这个枝条咯

- 好想成为暴力大师
- 朴素的 TLE 想法是 从低位往高位 BFS 枚举加 0/1 判断
- 但暴力大师看样例打表发现性质：一个稳数的任意后缀一定还是稳数
- 反过来，如果当前数不是稳数，就可以减掉这个枝条咯

- 实现细节，对于队列中的每个元素，设其十进制长度为 len，
- 保存其十进制下的哈希值和二进制下的最后 len 位
- 注意一下加入队列的顺序，先添 0 后添 1

- 至于“一个稳数的任意后缀也一定是稳数”这个结论的证明
- 有请我校教练的最强弟子 Krew 分享一下

- 最后安利一个构造题拖拖时间
- 构造一张无向图，点数边数随意
- 使得这张图中每个点的度都是 k，且至少存在一条边是桥
- $k \leq 100$，若无解输出 NO
- 桥：删除这条边后使图不再连通的边

- 先考虑有解条件
- 有解当且仅当 k 为奇数

- 证明
- 某个桥把原图分成两个子图，考虑每张子图，设其点数为 n
- 则整张子图的总度数为 $n \cdot k - 1$
- 由于是无向图，所以子图的总度数应为偶数
- 所以 n 和 k 都须是奇数
- Po 姐博客上还有个边双缩点的证明，很简单就不放了

- 为了节能，我们构造的两个子图当然要相同
- 由于每个点的度数为 k，所以一张子图至少要 $k+1$ 个点
- 但 $k+1$ 奇偶性不对，所以我们考虑 $k+2$ 个点
- 即现在已知度数数组，要求构造出原图
- 方法是每次找到剩余度数最大的点，设其度数为 d
- 往其他度数前 d 大的点连边即可

- 不确定这个套路的正确性怎么办呢?
- 我 yy 了一种乱搞
- 设三个特殊点 s, a, b，其中 s 为桥的端点
- 把 s 往剩下 $k-1$ 个普通点连边，s 就满足了条件
- 把 a 和 b 也分别往 $k-1$ 个普通点连边，然后 a，b 之间连边，a 和 b 也满足了条件
- 剩下 $k-1$ 个普通点，每个点的度数都是 3，都还差 $k-3$ 度
- 超强对称性，看成正 $k-1$ 边形，怎么搞都可以了
- 题不难，方法很多

EOF

- ありがとうございます ・ω・