Pilgrimage 解题报告

大连二十四中 于纪平

2013 年 9 月 27 日

1. 题目大意

有若干人参加了旅行，大家有共有财产，共有N个操作，分为 4 种：

(IN)加入k个人且每个人交纳原来平均每人的钱数；

(OUT)退出k个人同时每个人拿走自己的一份钱；

(COLLECT)从每个人收k元钱；

(PAY)从公共财产拿出k元使用。

已知在全过程中，所有的除法从未发生涉及分数的情况，求一切可能的初始人数。

数据规模：$N \le 50, k \le 2000, T$(每个文件的输入数据个数) ≤ 30000。

2. 算法分析

2.1. 问题转化

容易看出，本题中所有的 COLLECT 操作都是无用的。因为这不会对总钱数对人数的模数产生任何影响。第一次 IN/OUT 之前与最后一次 IN/OUT 之后的所有 PAY 操作也是无用的。而连续的几次 PAY 操作，相当于一次 PAY 操作，其k值为各次之和。所有的 OUT k操作均可以看做是 IN $-k$。

只有在 IN 操作时才会做除法，才有可能产生分数运算。所以对于每一次 IN 操作，我们可以得到一个信息：从上次 IN 之后到这次 IN 之前，所有的 PAY 操作的k值总和，除以上一次 IN 之后时当时的人数，余数为 0。

设初始人数为x，则对于每次 IN 操作，我们能够得到 1 个方程。形式为：$a \bmod (x + b) = 0$，其中a与b均为常数。事实上，a为从上次 IN 操作之后到现在的 PAY 操作的k值总和，b为在此之前所有 IN 操作的k值总和。总计有 $O(N)$个方程。

2.2. 算法 1：直接枚举答案

考虑x的范围。由于a最大可以为 $O(Nk)$，而x是的a约数，所以x也是 $O(Nk)$的。

枚举x依次代入检验，可以在 $O(N^2k)$的时间复杂度解决问题。特殊地，如果方程的

个数为 0，则应直接输出 SIZE $>=$ M，其中M为最小的满足全程都至少有 1 个人的初始人数。

2.3. 算法 2：对算法 1 的优化

实际上，我们所要求的是，对于 $\mathrm{O}(Nk)$范围内的每个x，分别满足了上述方程的多少个。显然我们对于以上的每个方程，可以求a的所有约数再减去b，直接找出所有符合该方程的x。

对于每个方程，我们需要用 $\mathrm{O}(\sqrt{k})$的时间处理，则对于N个方程，我们就可以在$\mathrm{O}(N\sqrt{k})$的时间内统计出每个x符合的方程个数，再从上一节所说的M到 $\mathrm{O}(Nk)$(x可能取到的最大值)枚举x扫一遍即可。总的时间复杂度为 $\mathrm{O}(Nk)$。

2.4. 算法 3：对算法 2 的优化

在上面的算法中，实际上我们对数组只进行了 $\mathrm{O}(N\sqrt{k})$次修改，所以，$\mathrm{O}(Nk)$扫一遍是没有必要的。我们需要用数据结构维护这个数组，例如 BST。（哈希表的遍历可能会有困难）

用 C++ STL 中的 map 可以很方便地实现，从而将其优化到 $\mathrm{O}(N\sqrt{k}\log N\sqrt{k})$。

3. 总结

本题是 ACM/ICPC World Finals 2006 的 Problem G。由于没有找到当时的测试数据，而在 uva 上提交简单模拟，也能够通过测试数据。然而，既然有稍微高效率，又不是很难的算法，我们当然需要更多的思考。由于制作测试数据不能扩大题目原本的数据范围，故只好采用了加大数据组数的方法。事实上，如果一次只测试一组测试数据，完全可以将N开到 1000，将k开到10^6，甚至更大。