

Prime Distance On Tree题解

雅礼中学 毛啸

2015 年 10 月 18 日

题意

假设当前有 n 个数形成的数组 $a[n]$ ，每次可以选择 v, t 两个数，满足 $v + t \leq n + 1$ ，且令 k 为最大的满足 $v + kt \leq n$ 的数，必须满足 $a[v] = a[v + t] = a[v + 2t] = \dots = a[v + kt]$ 。然后这些数将会删除(若 $v + t = n + 1$ 则只删除第 v 个数)，之后形成新的由 $n - k$ 个数形成的数组 $a[n - k]$ ，满足这个数组的元素是原来数组被删除之后剩下的元素，且相对位置保持不变。

设计一种方案删除所有的数，你的得分与你的方案的总步数有关。

对于50%的数据， $n \leq 100$ 。

对于100%的数据， $n, a[i] \leq 100000$ 。

题解

一个简单的方法是， i 从 n 到1输出 i 1，显然这样满足条件。

对于前一半的数据，可以采用小范围贪心搜索的办法，即每次搜索的时候，只从这些等差数列中搜，判断有限步（比如2步），能删除的最多的数，然后决定删哪一个比较好，显然搜索步数越多，方案越优。

对于后一半的数据，有一个办法是，找到出现次数最多的数，只保留这个数，其他数删掉，最后输出1 1。

这个算法有几个改进，有一个改进是，找到出现次数最多的数的最后一次出现，从那之后再找出现次数最多的数，每一个出现次数最多的数都一次消掉。

还有一个改进是，如果当前出现次数最多的数改变了，那么久一次把出现次数最多的数消掉。

这个算法无论基于什么思路，都是把序列分成很多个部分，每个部分保留出现次数最多的数。小数据很容易DP，大数据可以考虑用贪心或者其他非完美算法。

还有一个办法是，从后往前，对于只出现一次的数，直接删掉，对于出现多次的数，找到一个和当前数相同的数并把他们一起删掉（因为一个数再最后，显然可以删掉），用数据结构维护即可，CC上的数据直接暴力即可以通过。

一个简单的改进是如果这一对数按照这一个公差还可以向前扩展就继续扩展。

CC上的最优解做法不值得我们借鉴，因为他是通过套出CC上数据的规律，然后直接分情况讨论获得的解。不过他的勤奋还是很值得肯定的，700多行的代码，然而由于有一个人抄袭了他的代码，现在CC上有两个相同的程序得了最高分。

官方题解

CC 上的题解