NOIP 模拟赛
试题讨论与讲解
day2

Cydiater

清华大学

2019 年 9 月 2 日

1. Dove 爱旅游
   - 题意
   - 讨论
   - 题解
2. Cicada 爱烧烤
   - 题意
   - 讨论
   - 题解
3. Dove 的博弈
   - 题意
   - 讨论
   - 题解

题目大意

给定 n 个节点的树，每个节点有两种颜色，要求选择该树的一个子联通块，使得不同颜色的数量之差的绝对值最大。

数据范围

对于全部测试点，保证 $n \leq 10^6, a_i \in \{0, 1\}$。

测试点	n	特殊性质
$1, 2, 3, 4, 5$	≤ 20	
$6, 7, 8$	≤ 500	
$9, 10$	≤ 3000	保证 a_i 相同
$11, 12, 13$	≤ 3000	
$14, 15, 16, 17, 18, 19, 20$		

得分情况

算法一

$n \leq 20$

暴力枚举子联通块即可。
复杂度为 $O(n2^n)$，期望得分 $25pts$。

算法二

保证 a_i 相同

可以发现此时取整棵树作为答案即可，复杂度为 $O(n)$，期望得分 $10pts$，结合算法一期望得分 $35pts$。

算法三

$n \leq 3000$

设 $f(i,j)$ 表示在 i 内选择的子树大小为 j 时黑色点数量的最大值，设 $g(i,j)$ 表示在 i 内选择的子树大小为 j 时白色点数量的最大值。

容易发现转移是树上合并背包的形式，最后我们结合每个点的 f,g 值计算答案即可。

复杂度为 $O(n^2)$，期望得分 $65pts$。

算法四

标算

可以发现 j 的取值并没有实际意义，我们直接取 $f(i)$ 表示节点 i 为根的联通块白点减黑点的差值的最大值，$g(i)$ 表示差值的最大值。

转移很好进行，不在赘述，复杂度为 $O(n)$，期望得分 $100pts$。

题目大意

给定 n, m，询问有多少个字符集大小为 m 的字符串满足长度大于 1 的前缀中只有 n 为回文串。

数据范围

对于全部测试点，保证 $n \le 10^6, m \le 10^9$。

测试点	n	m
1,2,3,4	≤ 5	≤ 5
5,6,7,8	≤ 20	≤ 2
9,10,11,12	≤ 3000	≤ 3000
13,14,15,16	≤ 3000	$\le 10^9$
17,18,19,20	$\le 10^5$	$\le 10^5$
21,22,23	$\le 10^6$	$\le 10^6$
24,25		

得分情况

算法一

$n \le 5, m \le 5$

暴力枚举所有可能的字符串然后暴力判定即可。
复杂度 $O(m^n \times n^3)$，期望得分 $16pts$。

算法二

$n \leq 20, m \leq 2$

将暴力判定改为中回文串的判定使用 Manacher 或者哈希加速。
复杂度 $O(m^n \times n^2)$，期望得分 $32pts$。

算法三

$n \leq 3000$

加速判定肯定是没什么前途的，我们考虑一些别的方面。

对于一个长度为 n 的字符串来说，容易发现其中回文串的数量是 $m^{\lceil \frac{n}{2} \rceil}$。那么如果我们知道了不满足条件的字符串数量，也就能够计算满足条件的字符串数量。

考虑如何计算不满足条件的字符串的数量，也就是那些满足存在一个前缀同样是回文串的回文串。

算法三

$n \le 3000$

对于一个长度为 n 的回文串来说，如果其存在长度大于 $\frac{n}{2}$ 的前缀为回文串，那么一定存在长度小于 $\frac{n}{2}$ 的前缀为回文串。

证明很简单，我们只需要考虑截取回文前缀中越过回文中心的那个部分，考虑到这个前缀同样是回文串，那么我们将这个部分翻折到前缀同样也是回文串。然后转化为一个更短的前缀，这是同样的问题，我们继续执行，总能构造出一个长度不超过 $\frac{n}{2}$ 的回文前缀。

算法三

$n \le 3000$

考虑容斥原理，我们设 $f(i)$ 表示长度为 i 的满足条件的回文串数量，那么有

$$f(n) = m^{\lceil \frac{n}{2} \rceil} - \sum_{i < \frac{n}{4}} f(i) \times m^{\lceil \frac{n}{2} \rceil - 2 \times i}$$

复杂度为 $O(n^2)$，期望得分 $64pts$。

算法四

标算

发现这个递推式可以前缀和优化，直接优化即可。
复杂度为 $O(n)$，期望得分 $100pts$。

题目大意

给定一个长度为 n 的序列，第 i 个位置为 a_i，同时给定 m。每次一个位置上的数可以吃到相邻的权值不超过 $a_i + m$ 的值，对于每个数询问是否存在一种方案使得其可以存活。

数据范围

对于全部测试点，保证 $n \le 8 \times 10^6, a_i, m \le 10^9$。

测试点	n
1,2,3	≤ 20
4,5,6,7	≤ 3000
8,9,10	$\le 5 \times 10^4$
11,12,13	$\le 10^5$
14	$\le 5 \times 10^5$
15	$\le 10^6$
16,17,18,19,20	$\le 8 \times 10^6$

得分情况

算法一

$n \leq 20$

搜搜搜！
复杂度玄学，期望得分 $0pts - 15pts$。

算法二

$n \leq 3000$

容易发现，对于每个位置来说，其想保留到最后的最优策略一定是每次吃掉两边较小的那个，我们暴力模拟这儿过程即可。

复杂度 $O(n^2)$，期望得分 $35pts$。

算法三

$n \le 5 \times 10^5$

观察到对于每个数来说，如果采取最优策略，那么如果他不能留在最后了，一定存在一个区间，使得其可以吃掉区间内所有的数，但是两边的数都无法吃掉。

进一步可以发现，对于一个区间来说，如果存在一个数是卡在了这个区间内，那么区间内的所有数必定会被卡在这个区间内。

考虑到每个位置至多对应一个区间，因此我们只要能够找到所有的区间，然后通过差分覆盖，就能够确定所有不满足题意的数。

算法三

$n \le 5 \times 10^5$

对于一个数直接确定其左右端点是困难的，我们考虑对于每个左端点来确定其最远的右端点，对于每个右端点确定其可能的最远的左端点。

这个可以通过简单的二分得到。考虑对于一个左端点 l 来说，其可能的最远的右端点为 r，那么对于 $[l, r]$ 内的右端点对应的左端点来说，如果其对应的可能的最远的左端点在 l 之前，那么这两边就可以拼接成一个完整的区间。

因此对于每个左端点来说，我们找到满足条件的最远的右端点即可。这个可以利用一个 std::set 来实现。

复杂度 $O(n \log n)$，期望得分 $70pts - 75pts$。

算法四

标算

想要通过本题需要采用复杂度为 $O(n)$ 的算法。算法三需要使用二分与平衡树，难以优化。我们需要考虑别的思路。

对于第 i 个位置上的数来说，如果其是整体最大的，那么显然这个数可以留到最后。那么对于其他的数而言，我们找到其左边和右边第一个比他大的数，那么显然这个区间内的数，不管 m 的限制是多少他都可以吃掉。这个时候，如果左边或者右边的数可以留到最后，同时这个区间内的和加上 d 满足吃掉左右端点中的一个时的限制，我们也可以说这个数能够留到最后。

可以发现这个判断的依赖关系是不存在环的，我们只需要记忆化搜索判断一遍即可。而找左右两边第一个大于本身的数是经典问题，不在赘述。

复杂度为 $O(n)$，期望得分 $100pts$。