2022 年第四届湖北省大学生程序设计竞赛

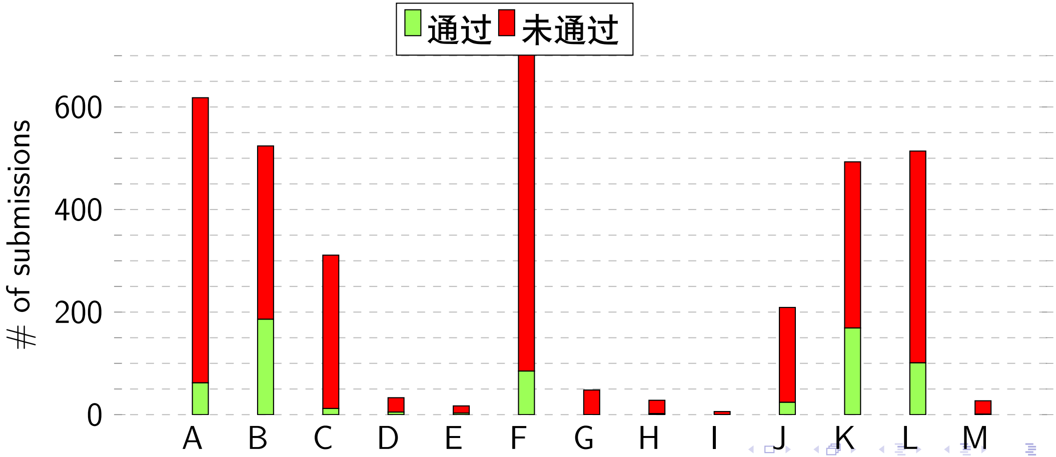
WHU ICPC/CCPC 集训队

Mar. 22nd, 2022

比赛小结

- 本次比赛共收到 2801 份提交代码。
- 其中 622 份代码正确。
- 166 个参赛队伍有提交记录。
- 166 个参赛队伍至少通过一题。
- 本次比赛一共有 13 道题目。
- 其中 11 道题目有队伍通过。

各题通过情况



A. Nucleic Acid Test

题意

欢迎大家参加 2022 年湖北省赛，参赛请先做核酸！
给定 n 个点，m 条边的无向图，其中有 k 个核酸检测点。要求从任意一个核酸检测点出发，两次核酸之间不能超过 t 分钟，问若要遍历图上所有点，并在某个核酸点结束，你的速度最小是多少。
$n \leqslant 300, t \leqslant 10^9$，图不一定连通。

A. Nucleic Acid Test

分析

$n \leqslant 300$，先考虑 Floyd 得到所有点对之间的距离。对任意一个点 x，如果到核酸点 A 和 B 的距离分别为 d_A, d_B，如果从 A 出发经由 x 到 B，用时为 $d_A + d_B$，造成的影响是 x 被访问。

不妨设 $d_A > d_B$，为了让 x 被访问，只需要从 B 出发，走到 x 再回到 B，就可以得到一种用时更短的方法。

（考虑到武汉最近核酸情况，没事就去做个扩面核酸续时间也是很容易理解的）

A. Nucleic Acid Test

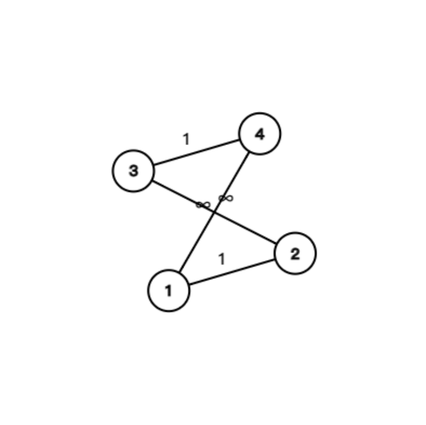
分析

因为题目要求访问所有点，那么可以将核酸点和非核酸点分开考虑。
非核酸点**做完核酸并回到一个核酸点**对答案的贡献一定是它到最近核酸点距离的 2 倍。而如果只考虑非核酸点，很有可能让整个图分成几个独立的部分。
因此核酸点之间要通过最小生成树来连接，对答案的贡献是最小生成树中最大的那条边。

A. Nucleic Acid Test

Corner

图不连通（不连通的可以是核酸点或非核酸点）。



$T = 0$

A. Nucleic Acid Test

题解

首先用 Floyd 将任意两个点之间的最短路求出来，然后对核酸点之间跑最小生成树。

记录最大的 $longestpath$，非核酸点的贡献是它到最近核酸点距离的 2 倍。

核酸点的贡献是最小生成树中最大的那条边。

注意到经过一条边的时间为 $\frac{l}{v}$ 不取整，可知本题本质为求出最大贡献后整体计算所需速度。

答案为 $\left\lceil \frac{longestpath}{t} \right\rceil$。

时间复杂度 $O(n^3)$。

该做法不必二分，但验题人基本都写了二分。

A. Nucleic Acid Test

总结

本题定位为 Easy-medium，只要求会写 Floyd，然后可以用各种方法通过本题。
这道题，无疑是善良的出题人无私的馈赠。大量精心构造的测试数据，涵盖了严谨而不毒瘤的 corner cases。你可以利用这道题，给你的比赛一个完美的开始。不多的数据组数、不大的数据范围和多种多样的数据类型，能让程序中的错误无处遁形。出题人相信，这个美妙的题目，可以给拼搏于 AK 这场比赛的逐梦之路上的你，提供一个有力的援助。

B. Potion (easy version)

题意

有一个容量仅在一半位置有刻度的量杯，有两类水，求最小接水步数使得杯子里面两类水的比例为 $x:y$，或者输出无解。

题解

假设当前第一类水的比例为 d ，那么一次接水后其比例变为 $\frac{d}{2}$ 或 $\frac{d+1}{2}$，初始时 $d=0$ 或 1。

分析一下可以发现，如果把 d 化为最简分数，那么分子总是奇数，分母总是 2^n，且 $n+1$ 是总接水次数，其他的打表找规律同样可以得到相同结论。

对 x 与 y 约分后判断它们是否为奇数且它们的和是否为 2 的次幂即可。

关于约分正确性的证明见 hard version。

C .Potion (hard version)

题意

有一个容量 $a+b$ 且仅有一个刻度 a （从底部往上）的量杯，有两类水，求最小接水步数使得杯子里面两类水的比例为 $x:y$，或者输出无解。

题解

为了方便讨论，我们先钦定接水次数为 $n+1$ ，其中第一次接水表示从 0 接满，其余接水表示从 a 接满，且 a,b 与 x,y 是分别约分之后的结果。

设 $c=a+b$ 那么第 $i(i>1)$ 次接到的水在 $n+1$ 次后所占的比例为 $\frac{b}{c}(\frac{a}{c})^{n+1-i}=\frac{ba^{n+1-i}}{c^{n+2-i}}$，第一次接水所占的比例为 $1\times(\frac{a}{c})^{n}$，对这些数进行通分，可以得到 $\frac{a^n+\sum_{i=1}^{n}ba^{n-i}c^{i-1}}{c^n}$.

C .Potion (hard version)

题解

考虑第 i 次接的是什么水，那么若 $x+y=c^n$ 且 x 等于集合 $\{a^n, ba^{n-1}c^0, ba^{n-2}c^1, \ldots, ba^0c^{n-1}\}$ 的某个子集的元素和，且 y 等于此子集补集的和，那么就有解且最小次数为 $n+1$.

对 a^n 是否存在进行讨论，问题转化为查询 $s=\sum_{i=1}^{n} a^{n-i}c^{i-1}\delta_i$ 是否存在，其中 $\delta_i \in \{0,1\}$ 表示第 i 项是否存在，设 $f(s,n)$ 表示相应的解。

我们逐项考虑是否存在，若 s 被 c 整除，由 a,c 互质可知项 $a^{n-1}c^0$ 不存在，转入子问题 $f(s/c, n-1)$，否则若有解一定存在项 $a^{n-1}c^0$，转入子问题 $f((s-a^{n-1})/c, n-1)$.
递归求解即可，复杂度 $O(T\log V)$

C .Potion (hard version)

题解

下面对约分的正确性进行证明：

a, b 约分是 trivial 的，不多赘述。

考虑未约分 x, y ，它们存在最小非平凡公因数 $d > 1$ 。

如果 x, y 无解，一定去约分尝试能不能变得有解。

否则有 $x + y = c^n$ 且 $d|c$ ，注意集合中仅有 a^n, ba^{n-1} 这两项不被 d 整除，且 $d|ca^{n-1}, ca^{n-1} = a^n + ba^{n-1}$，故这两项必须同时属于 x 或 y，故此时一定有 $c|x \wedge c|y$.

由上可知，若 x, y 不互质，一定有 $\gcd(x, y) = c^p$，此时分子分母同时约去这些因子 c ，一定能让步数减少。

D. Transition

题意

有两个操作，一是反转某位，代价为 1，另一个是交换两个位置，代价是两者位置绝对值之差。求把 01 序列 a 变成 b 的最小代价下的操作集合数量。

D. Transition

题解

先考虑最小代价，交换相邻两位的平均单位改变代价为 $\frac{1}{2}$，反转单位的平均单位改变代价为 1，这是两个平均改变代价最小的操作。

以上启发我们，交换操作的位置差必须很小，具体的，交换仅可能和前两位发生，交换与翻转不会重叠，且相邻交换会不会连续发生超过两次（即不可能发生 $(i, i+1), (i+1, i+2), (i+2, i+3), \ldots$ 的情况）并只存在一种顺序。

那么直接对二元组 (代价, 数量) 进行 dp 即可，每次讨论前两位的情况进行转移。

D. Transition

题解

下面给出证明:
称 swapa 表示一次位置差为 a 的交换，flip 表示一次翻转，两个操作重叠指它们同时作用了至少一个位置。

讨论最优集合中可能出现的操作。

1. 假设最优集合中存在 swap3，那么其一定可以被两个 flip 代替使得代价更小，故最优集合不存在 swap3.
2. 假设最优集合中存在 swapa 与 flip 重叠的情况，那么其一定可以被一个 flip 代替使变得更小，故最优集合中不存在这两种操作重叠情况。
3. 假设最优集合中存在 swap2 与 swapa 重叠的情况，由于 swap2 一定可以被两个 flip 等价替换，那么被规约到第二条，可知不存在。

D. Transition

题解

4. 最后我们考虑 swap1 内部的重叠情况，假设重叠部分长度为 $n(n > 2)$，由于全 flip 代价仅有 n，因此这部分重叠的 swap1 个数不大于 n，同时由于需要处处重叠，因此个数不小于 $n-1$ ，仅有两种情况。

○ swap1 个数为 $n-1$ ，注意到 swapa 操作必须作用在一对 0 和 1 时才是有效操作，那么除了左右端点位置以外，所有其余位置被 swap1 作用了两次，于是这一连串操作等价于交换左右端点，可以规约到第一条，即 $n-1 \leqslant 2$ ，重叠部分只能有 2 个 swap1.

○ swap1 个数为 n ，此时若合法，一定至少需要全部 flip，即 n 个位置全部与目标不同，又因为 swap1 不改变 01 的数量，因此 n 个位置中必定同时存在 01，那么对一个 01 相邻的位置使用 swap1 ，其余位置使用 flip 后，代价为 $n-1$，因此此情况不存在。

E. Multigate

题意

给两个数组 a_i, b_i 和一个正整数 x_0，有递推 $x_i = \begin{cases} x_{i-1} \text{ and } a_i & b_i = 0 \\ x_{i-1} \text{ or } a_i & b_i = 1 \end{cases}$

T 次独立询问 x_0, k，问可以反转至多 k 个 b_i 后最大的 x_n.

题解

反转最后 k 个 $b_i = 0$ 的位置一定是最优的，独立考虑每一位后，可以分别维护 0,1 经过前缀后与经过有翻转后缀后的结果，每次直接找到中间点进行查询即可，复杂度 $O(30n)$.

E. Multigate

题解

也可以不独立考虑每一位，预处理后缀影响 $c_{n+1} = 0, c_i = c_{i+1} \text{ or } a_i$，预处理前缀影响 $y_0 = 2^{30} - 1, y_i = \begin{cases} y_{i-1} \text{ and } a_i & b_i = 0 \\ y_{i-1} & b_i = 1 \end{cases}$，令 x_i 表示 $x_0 = 0$ 时 x 的递推结果，那么假设目前询问处在位置 i，询问初始值为 v，答案即为 $v \text{ and } y_i \text{ or } x_i \text{ or } c_{i+1}$，复杂度 $O(n+q)$.

下面给出最后 k 个一定优的证明：
考虑两个 0 位 $i < j$，如果我们翻转了 i 位置而没有翻转 j，那么分情况考虑它们俩二进制某一位，如果 j 这一位为 0，那么在不翻转 j 位置时翻 i 位置是没有意义的；如果 j 这位为 1，那么翻转 i 一定没有翻转 j 优秀。

F. Angel

题意

一排有 n 个洞，有一只兔子在某个洞，每个时刻必定移动到相邻的洞中，需要构造长度最短的询问序列 q_i，第 i 项表示询问在兔子第 i 次移动前是否在 q_i 这个洞中，使得至少猜中一次。

解法

思路一：考虑兔子发生移动时奇偶性必定发生改变，那么可以钦定兔子初始奇偶性。确定了奇偶性后，由于兔子一次只能走一步，所以可以把它往一边赶。最坏情况下重复两遍这个过程即可，对边界稍加讨论可以发现 $2(n-2)$ 大概就是这个思路下的最小值，注意特判 $n = 1, 2$ 的情况。

F. Angel

解法

思路二：使用状压 DP 打表找规律，令 dp_s 表示兔子可能存在的状态为 s 的最小询问数，枚举询问地点进行转移即可。

还有许多其他思路，例如建立兔子的时间-位置坐标等等。

接下来给出 $2(n-2)$ 是最小值的证明：

F. Angel

解法

证明：由思路一，当 $n > 2$ 时，初始时刻在奇数位置情况的兔子和在偶数位置情况的兔子永远不会相交，且每一时刻后奇偶互换。
考虑一次检查实际上是对某一类奇偶性消除了一个可能，可以发现，如果两类奇偶性都存在且存在空位，那么检查一类奇偶性会使得另一类奇偶性增加一个可能性，所以我们必须把一种奇偶性先全部消除。
消除一种奇偶性情况时，另一类奇偶性会变成此类奇偶性，重复操作即可。可以发现，消除一类奇偶性至少需要 $n-2$ 步，则总步数至少需要 $2(n-2)$ 步。

Bonus：此题有树上版本，详见：
https://math.stackexchange.com/questions/4418051/how-do-you-catch-a-cat-on-a-tree

G. Brick

题意

给一个正整数数组 h_i，每次可以把一个 h_i 加 2 或者把两个相等的 h_i, h_{i+1} 同时加 1，问使得所有 h_i 相等的最小高度或者输出无解。

G. Brick

题解

题解一：

按照 h_i 的奇偶性对每个位置分类，那么竖着放不改变奇偶性，横着放改变两个相邻位置的奇偶性，最终要求所有位置的奇偶性相同。

考虑奇偶性相同的某两个位置 i, j ，如果想通过横放改变它们的奇偶性，一个直观的想法是，如果它们之间所有元素的奇偶性都与它们不同，且它们之间有偶数个位置（或者是，它们位置的奇偶性不同），即可做到。

对比中间位置最高高度与这两个端点之间的高度，可以分为两类情况 $100001 \rightarrow 111111$ 和 $011110 \rightarrow 211112 \rightarrow 222222$ ，注意这里忽略了对矮的位置补充竖着的砖块的过程。

G. Brick

题解

更进一步的，把位置分为奇高度位置和偶高度位置两种，如果存在至少一种高度位置满足，奇数位置个数等于偶数位置个数，就是有解的。

接着，对于有解的情况，我们枚举是把所有奇高度位置变成偶高度位置还是反过来。考虑这个转化过程，对于某种高度的位置集合，如果我们把奇数位置视为'('，偶数位置视为')'（或者反过来），我们发现这必须是一个就近的括号匹配的过程，每次括号匹配是区间改变高度的过程，按上述两类情况，用差分维护即可，复杂度 $O(n)$.

G. Brick

题解

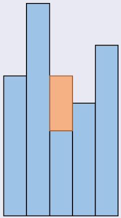
题解二：

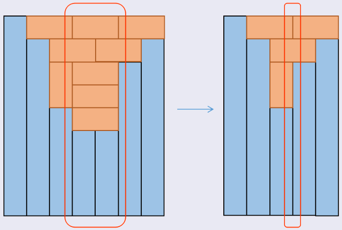
一个简单的 $O(nh)$ 做法：

考虑 h_i 最小的任一个点 i ，

1. 如果 $h_{i-1} > h_i$，$h_i < h_{i+1}$ ，只能在点 i 填上竖向的方块
2. 如果 $h_i = h_{i+1}$ 可以从 $\{h_n\}$ 中同时删去 i 和 $i+1$ 两点，答案不变

重复以上过程





G. Brick

题解

链表 $+$ 优先队列维护“在哪个高度可以删去相邻的这两个位置”就可以做到 $O(n \log n)$

H. Hamster and multiplication

题意

定义函数 $f(x)$ 如下，求解 $\sum\limits_{i=1}^{n} f(i)$。

$$f(x) = \begin{cases} x & (x < 10) \\ f(\prod_i x_i) & (x \geqslant 10) \end{cases}$$

题解

注意到 $x < 10^{18}$ 时，$\prod_i x_i$ 的可能取值其实只有 29242 种。
因此用 `std::map` 之类的数据结构，可以维护 `dp[bit][res]` 数组，表示数有 bit 位时，$\prod_i x_i = res$ 的 x 的数量。

H. Hamster and multiplication

题解

令 $n = n + 1$，下面统计的是 $\sum_{i=1}^{n-1} f(i)$ 的答案。

假设 n 的最高位为 t，x 的第 i 位是 x_i（从低到高），答案可以按照如下方案进行统计。（如果你做过数位 DP 的话，应该就好理解了）

统计所有数位小于 t 的数，也就是：$\sum_{i=1}^{t-1} \sum^{res} dp[i][res] \times f(res)$。

统计位数为 t，小于 n 的数。从 x_t 开始枚举倒序 x_i，记 $vtop_i = \prod_{j=i}^{t} x_i$，

那么这一部分的结果可以表示为：

$$\sum_{0}^{i=t} \sum_{j=0}^{x_i - 1} \sum^{res} dp[i][res] \times f(res \times j \times vtop_{i+1})$$

I. Latitude Compressor

题意

求所有 n 阶排列的 m 次方有多少种不同的取值，即求 $|\{p^m \mid p \in \text{permutation}(n)\}|$。

题解

需要知道的性质：

- 置换可以唯一分解成若干个独立的环
- 如果 p 是一个大小为 a 的环，那么 p^m 分解后是 $\gcd(a,m)$ 个大小为 $\frac{a}{\gcd(a,m)}$ 个环

I. Latitude Compressor

题解

考虑 p^m 由哪些环组成：定义 bool 数组
$b_{x,y} \in \{0,1\}$，$(1 \leqslant x \leqslant n, 0 \leqslant y \leqslant \lfloor \frac{n}{x} \rfloor)$ 表示在置换
$q \in \{p^m \mid p \in \mathrm{permutation}(n)\}$ 中，是否可以出现恰好 y 个大小为 x 的环

$$b_{x,y} = 1 \iff \exists a_1, a_2, \ldots a_l \text{ 满足：}$$

$$\forall a_i, \frac{a_i}{\gcd(a_i, m)} = x$$

$$\sum_i \gcd(a_i, m) = y$$

通过简单背包 dp 可以求得 $b_{x,y}$，也可以进一步观察性质发现：
$b_{x,y} = [t_x \mid y]$ 其中 $t_x = \min\{d \mid \gcd(x, \frac{m}{d}) = 1\}$

题解

现在，可以通过 $b_{x,y}$ 写出答案的指数型生成函数 (EGF)：

$$\text{answer} = n![x^n] \prod_{i=1}^{n} \sum_{j\geqslant 0} b_{i,j} \frac{\left[(i-1)!\frac{x^i}{i!}\right]^j}{j!}$$
$$= n![x^n] \prod_{i=1}^{n} \sum_{j\geqslant 0} b_{i,j} \frac{x^{ij}}{i^j \cdot j!}$$

一种思考方式：单个有标号、大小为 i 的环的 EGF 为 $f = (i-1)!\frac{x^i}{i!}$，多个有标号、相同大小的环的 EGF 为 $\exp(f) = 1 + \frac{f}{1!} + \frac{f^2}{2!} + \dots$ （SET 构造），考虑到 $b_{x,y}$ 的限制，稍作修改即可

题解

这个式子不能直接计算，但可以利用多项式 ln / exp 的技巧求出：

定义 $F_i(x) = b_{i,j}\ \frac{x^j}{j!}$，

$$\text{answer} = n![x^n]\ \prod_{i=1}^{n}\sum_{j\geqslant 0} b_{i,j}\ \frac{x^{ij}}{i^j \cdot j!} = n![x^n]\ \prod_{i=1}^{n} F_i(\frac{x^i}{i})$$

$$= n![x^n]\ \exp(\sum_{i=1}^{n} \ln F_i(\frac{x^i}{i}))$$

注意到 $\deg F_i(x) \leqslant \lfloor \frac{n}{i} \rfloor$，可以 $O(n\log^2 n)$ 求出所有的 $\ln F_i(x)$，通过修改下标得到 $\ln F_i(x^i)$

(ln 的本质是多项式变换，$\ln(1+f) = 1 - f + \frac{f^2}{2} - \ldots$，因此可以换元)，

复杂度：$O(n\log^2 n)$

I. Latitude Compressor

题解

更优秀的做法：如果注意到了 $b_{x,y} = [t_x \mid y]$，定义 $G_t(x) = \sum_{j\geqslant 0} \frac{x^j}{(jt)!}$

$$\text{answer} = n![x^n] \prod_{i=1}^{n} \sum_{j\geqslant 0} [t_i \mid j] \left(\frac{x^i}{i}\right)^j \frac{1}{j!}$$

$$= n![x^n] \prod_{i=1}^{n} \sum_{j\geqslant 0} \left(\frac{x^i}{i}\right)^{jt_i} \frac{1}{(jt_i)!} = n![x^n] \prod_{i=1}^{n} G_{t_i}\left(\frac{x^{it_i}}{i^{t_i}}\right)$$

对相同的 t 一起处理，可以证明复杂度：（下文中 f 表示因子次数）

$$\sum_t \frac{n}{t\cdot(\min\{i \mid t_i = t\})} \log n = \sum_t \frac{n}{\left(p_1^{f_{p_1}(m)} p_2^{f_{p_2}(m)} \dots p_l^{f_{p_l}(m)}\right) \cdot (p_1 p_2 \dots p_l)} \log n$$

$$\leqslant \sum_t \frac{n}{(p_1 p_2 \dots p_l)^2} \log n \leqslant \sum_{i\leqslant n} \frac{n}{i^2} \log n = O(n \log n)$$

J. Palindrome Reversion

题意

给一个字符串 s，问能否翻转 s 的一个区间使 s 回文。

解法

设 $|s| = n$，可以发现这样一个性质：

如果 $s[1] = s[n]$，并且可以翻转 $s[1, i]$ 使 s 回文，
那么有 $s[i] = s[n] = s[1]$，翻转 $s[2, i-1]$ 同样可以使 s 回文。

两边先删去相同字符，之后只需判断 s 是否为 AAP, PAA, APA 的形式即可（A 为任意串，P 为回文串），最简单的做法是字符串哈希。

K. PTT

题意

给 T 组 Arcaea 的游玩成绩和谱面定数，求每次游玩的单曲 PTT

题解

按题目给的公式算就行，注意单曲 PTT 为非负数。
数据范围也没有什么坑，也不会卡精度什么的，是一道非常良心的签到题。
一开始其实打算把 best30 和 recent10 加上出的复杂一点的，但是考虑到样例会变长好多就放弃了。
~~最后欢迎添加出题人的 Arcaea 好友 481198937~~

L. Chtholly and the Broken Chronograph

题意

给出一个数列 $a_1, a_2, \ldots, a_n$，要求支持以下 4 种操作：

1. 禁用位置 x
2. 启用位置 x
3. 对区间 $[l, r]$ 中启用的位置加 x
4. 查询区间 $[l, r]$ 的和，不论每个位置状态如何

题解

本题并不需要使用非常复杂的数据结构，仅考察了线段树的基本操作。
在线段树上的每个节点记录当前区间的和与启用状态的位置个数。
1, 2 操作单点修改状态，3 操作仅对启用位置更新和，4 操作正常查询即可。

M. Super Star Spectacle

题意

收集器（华恋）在树的路径上巡逻，问确认树上没有很会润的 Star 最少需要几个收集器

解法 1

如果指定一个巡逻的起点作为树的根，在有根树上会有如下的 DP：
记 $f(i)$ 为以 i 为根的子树的最优解，考虑 i 的所有儿子 j，

- 如果 $f(j)$ 的最大值 m 只出现了一次，那么可以驻守一收集器在 i，以 $m-1$ 收集器巡逻其他子树，最后一起巡逻 m 的子树，于是 $f(i)=m$
- 如果 m 不止出现一次，那么需要额外驻守一收集器在 i，$f(i)=m+1$

考虑换根 DP，维护最大值、次大值及其出现次数，反转树边（删去子对父的贡献，计算父对子的贡献），即可求出以所有点为起点的答案，$O(n)$

M. Super Star Spectacle

解法 2

虽然本题的换根 DP 不难写，也不要急着写，观察换根 DP 中 $f(i)$ 的性质，或者直接考虑出并喜欢上以下结论：

- 右图中在同一层中的点与边可以用同一收集器巡逻

考虑怎么划分出这些层：每一轮删掉叶子与相连的链，直至只剩下一条链。使用类似拓扑排序的做法，好写，$O(n)$