

SHCPC2025 题解

SHCPC2025 出题组

2025 年 7 月 5 日

序列

定义一个序列的权值为所有非空子序列的颜色数之和。

求这个权值的方式是，记每种颜色 c 出现的次数为 cnt_c ，则权值为

$$\sum_{c=1}^n 2^{n-cnt_c} \times (2^{cnt_c} - 1)$$

序列长度为 n ，则权值处于 $[2^n - 1, n \times 2^{n-1}]$ 之间。

因此，可能成为答案的 n 的种类是 $O(\log \log x)$ 的。

序列

对一个 n ，本质不同的序列种类只有划分数 $p(n)$ 种。

因此，我们只需要枚举可能的 n ，然后进行搜索判断即可，需要进行完善的剪枝。

时间复杂度为 $O(p(\log x) \log \log x)$ ，其中 $p(60) = 966467$ 。

审判

以下认为 n, M 同阶。

注意到：如果当前处决指数为 x ，可以将所有 b_i 换作 $\frac{b_i}{x}$ ，可以化归到处决指数为 1 的情况。从而可以发现题目所述的过程有很强的递推关系。

设 $f(a_1, a_2, \dots, a_k)$ 为有 a_i 次第 i 类攻击时的暴力等级，所以我们可以得到如下的递推式：

$$f(a_1, a_2, \dots, a_k) = \max_{b_1 + \dots + b_k = 0, b_i \geq -1} \left\{ \min_{1 \leq j \leq k} \{ (1 + b_j) f(\dots, a_j - 1, \dots) \} \right\}$$

审判

由这个递推式我们可以得到：

$$f(a_1, a_2, \dots, a_k) \leq (1 + b_j) f(\dots, a_j - 1, \dots)$$

$$\frac{1}{f(\dots, a_j - 1, \dots)} \leq \frac{1 + b_j}{f(a_1, a_2, \dots, a_k)}$$

对 $j = 1, 2, \dots, k$ 累和：

$$\sum_{j=1}^k \frac{1}{f(\dots, a_j - 1, \dots)} \leq \frac{k}{f(a_1, a_2, \dots, a_k)}$$

审判

而我们对所有可能的 $(b_1, b_2, \dots, b_k)$ 取 $\max$, 并且上式取等号时反解即可得到一组合法的 $(b_1, b_2, \dots, b_k)$, 因此:

$$\sum_{j=1}^k \frac{1}{f(\dots, a_j - 1, \dots)} = \frac{k}{f(a_1, a_2, \dots, a_k)}$$

简单来说, 就是 $f(a_1, a_2, \dots, a_k)$ 等于 $f(\dots, a_j - 1, \dots)$ 的调和平均。

设 $g(a_1, a_2, \dots, a_k) = \frac{k^{a_1 + \dots + a_k}}{f(a_1, a_2, \dots, a_k)}$, 那么我们有递推式:

$$g(a_1, a_2, \dots, a_k) = \sum_{j=1}^k g(\dots, a_j - 1, \dots)$$

审判

不难发现这个递推式就类似于 k 维空间中的杨辉三角，也就是从 $(0, 0, \dots, 0)$ 走到 $(a_1, a_2, \dots, a_k)$ 的最短路径方案数，故：

$$g(a_1, a_2, \dots, a_k) = \binom{a_1 + a_2 + \dots + a_k}{a_1, a_2, \dots, a_k}$$

所以我们得到了 $f(a_1, a_2, \dots, a_k)$ 的通项公式：

$$f(a_1, a_2, \dots, a_k) = \frac{k^{a_1 + \dots + a_k}}{\binom{a_1 + \dots + a_k}{a_1, \dots, a_k}}$$

审判

这时我们就可以代回所求的式子进行计算：

$$\sum_{T=0}^M \sum_{a_1+\dots+a_k=T, a_i \leq n} f(a_1, a_2, \dots, a_k) = \sum_{T=0}^M \frac{k^T}{T!} \sum_{a_1+\dots+a_k=T, a_i \leq n} a_1! a_2! \dots a_k!$$

设 $h(x) = \sum_{i=0}^n i! x^i$ ，那么最后那一项就是 $[x^T] h(x)^k$ ，所以用多项式快速幂即可。

时间复杂度： $O(n \log n)$ 。

饺子

首先由于 $b_i > 0, c_i \geq 0$, 所以 $e_{i,j}$ 是单调递减的。

也就是说, 定义 $f(x)$ 为吃了 x 个饺子的最大愉悦值, 不包含额外愉悦值, 则 $f(x)$ 是一个上凸函数。

如果不考虑额外愉悦值, 那么答案只可能在 $0, m$ 和函数极值点取得。

考虑额外愉悦值的情况, 答案可能的取值点只可能增加 l, r 两个点。

对每个取值点进行二分求解即可, 复杂度 $O(n \log v)$ 。

与或博弈

为了简化问题，我们把 b, y 按位取反，对 b 的操作同时进行变化，即：

先手操作：

- $a := a \& v$
- $b := b \& v$

后手操作：

- $a := a | v$
- $b := b | v$

容易证明变换的等价性。

与或博弈

先只考虑 a ，如果当前 $a \& x \neq x$ ，则 a 永远无法变成 x 。

证明：先手对 a 进行一步变换只能使 a 变成 a 的子集，而 x 不是 a 的子集，此时后手只要把 a 变回原来的值，先手就无法获胜。

因此，如果 $a \neq x$ 且 $b \neq y$ ，则先手必败。

当且仅当 $a = x$ 和 $b = y$ 至少成立一个，且另一个可以一步达到目标，此时先手才能获胜。

Djangle 的数据结构

线段树，维护区间和、区间 gcd、区间 lcm

对于操作 1：直接维护即可

对于操作 2：可以看成先对区间做 $x \rightarrow \gcd(x, a[i])$ 然后求区间和

对于 $\text{lcm} \mid x$ 的节点，本次操作不会改变该节点，可直接跳过

否则修改暴力递归左右区间

打标记的条件：当前节点 $\text{gcd} = \text{lcm}$

Djangle 的数据结构

复杂度分析:

定义总势能为: 线段树全相等节点 (且父节点不全相等) 的 $\log \text{val}$ 的总和

每次操作 1 最多修改 $\log n$ 个节点, 势能至多增加 $\log^2$

操作 2 最多拆开两侧两个全相等的区间, 相当于做了两次操作 1, 对于中间的部分, 每遍历一个区间, 势能至少降低 1, 这样的暴力最多做 $O(n \log^2 n)$ 次

$O(n \log^3 n)$, 三个 $\log$ 分别是: 线段树、操作 1 增长势能、gcd/lcm 的 $\log$

No explanation

预处理字符串 i 从 `luolikong` 的第 j 个位置开始可以匹配到哪个位置，记录为 $\text{nxt}_{i,j}$ 。

记当前状态 s 为目前 `luolikong` 最长匹配到了哪个位置，不难发现 $s \in [0, 8]$ 。

我们把经过的状态进行状态压缩，即所有的字符串总共经过了哪些状态，记为 S 。

例如，如果连接字符串 `luo`、`lik`，则经过了状态 $0, 3, 6$ ， $S = 100100100$ 。

No explanation

枚举答案经过了哪些状态 S ，对每个 S 进行求解。

把字符串连接看成在图上的一条路径，则需要使得路径上每条非自环边都有一个字符串匹配。

首先如果一个字符串存在 $\text{nxt}_{i,j} = j$ ，且 $j \in S$ ，则将这个字符串加入答案，即取了一个自环。

接下来进行状压 dp，设 dp_t 表示当前匹配了集合 t 的连边的最大字符串长度。（这里压缩了一维状态：用到了前 i 个字符串）

转移时，若当前字符串已经加入自环集合，则贡献为 0；否则，贡献为字符串长度。

令 $k = 8$ ，则复杂度为 $O(k3^k n + k \sum |s_i|)$ 。

No explanation

验题人给出了一种和上面做法前半部分相同的做法：枚举每个状态，之后用最大费用最大流进行二分图最大权匹配。其中二分图左边只有 k 个节点。

矩阵

设大于 n 的第一个质数为 prime
构造方案如下:

$1, 2, 3, \dots, n$

$1 + \text{prime}, 2 + \text{prime}, 3 + \text{prime}, \dots, n + \text{prime}$

$1 + 2\text{prime}, 2 + 2\text{prime}, 3 + 2\text{prime}, \dots, n + 2\text{prime}$

$\vdots$

$1 + (n - 1)\text{prime}, 2 + (n - 1)\text{prime}, \dots, n + (n - 1)\text{prime}$

打表可知, 1 到 2500 内两个素数的差不超过 40
该构造方式显然满足条件
时间复杂度: $O(n^2)$

V 我 112.5

签到题

输出 $50 \times (\frac{x}{100} + 1)$ 即可

时间复杂度: $O(1)$

真相

注意到： i 号节点逻辑自洽的条件是

- 如果 i 是诚实者，那么 i 子树 (不包含 i) 内恰好有 $a_i - 1$ 个人诚实者
- 如果 i 不是是诚实者，那么 i 子树内 (不包含 i) 不能恰好有 a_i 个人诚实者

令 $dp_{i,j}$ 表示子树 i 中恰好有 j 个诚实者的方案数，然后做类似于树上背包的转移即可。

设 $f_{i,j}$ 表示子树 i （不包含节点 i ）中恰好有 j 个诚实者的方案数，则有：

$$f_{i,j} = \sum_{k=0}^j f_{i,k} \times dp_{son,j-k}$$

对于 $dp_{i,j}$ ，有：

$$dp_{i,j} = \begin{cases} dp_{i,j} = f_{i,j-1} & j = a_i \\ dp_{i,j} = f_{i,j} & j \neq a_i \end{cases}$$

最后答案为

$$\sum_{i=0}^n dp_{1,i}$$

时间复杂度 $O(n^2)$ 。

画卷

把操作看成两种情况：

- 把一个任意白边变黑，贡献 +0
- 把一个两端被黑边连通的白边变黑，贡献 +1

按照先黑边后白边的顺序并查集维护即可

时间复杂度： $O(m\alpha(n))$ ，其中 α 是反阿克曼函数

神之一手

本题思路不难，细节偏多。

由于界外棋子 ≥ 2 即输掉，所以我们只需要维护当前界外棋子为 0, 1 的概率，和当前提过多少棋子。

每个回合开始时，先把当前界外棋子为 1 的概率乘上 r ，加到答案里，然后对剩下的概率进行计算：

把棋子分成 80 之前和 80 之后两部分，掉出棋盒盖的概率不同，相乘即可求出当前 a_i 个棋子中，总共掉出 0 个棋子，1 个棋子的概率。

时间复杂度 $O(n \log P)$ 。

神之一手

具体地，设 p_0 为当前界外棋子为 0 的概率， p_1 为当前界外棋子为 1 的概率， sum 为前 $i-1$ 个回合提过的总棋子数。

首先先将 $p_1 r$ 加到 ans_i 中，之后 $p_1 := p_1(1-r)$ 。

计算 a_i 个棋子全部放入棋盒盖的概率 pp_0 ：

$$pp_0 = \begin{cases} (1-p)^{a_i} & sum + a_i \leq 80 \\ ((1-p)(1-q))^{a_i} & sum \geq 80 \\ (1-p)^{80-sum}((1-p)(1-q))^{a_i+sum-80} & otherwise \end{cases}$$

神之一手

计算恰有一个棋子掉出棋盒盖的概率 pp_1 :

当 $sum + a_i \leq 80$ 时: $pp_1 = a_i \cdot p \cdot (1 - p)^{a_i - 1}$

当 $sum \geq 80$ 时:

$$pp_1 = a_i \cdot (1 - (1 - p)(1 - q)) \cdot ((1 - p)(1 - q))^{a_i - 1}$$

否则: 分两部分计算

$$\begin{aligned} pp_1 = & (80 - sum) \cdot p \cdot (1 - p)^{79 - sum} \cdot ((1 - p)(1 - q))^{a_i + sum - 80} \\ & + (1 - p)^{80 - sum} \cdot (a_i + sum - 80) \cdot (1 - (1 - p)(1 - q)) \\ & \cdot ((1 - p)(1 - q))^{a_i + sum - 81} \end{aligned}$$

最后更新: $p'_0 = p_0 \cdot pp_0$, $p'_1 = p_0 \cdot pp_1 + p_1 \cdot pp_0$ 。

迷宫

建图方式：

将一个点拆成上下左右四个点，相邻两个反弹板公用一个节

点

对于 \ 形式反弹板：连接上右两个点，左下两个点

对于 / 形式反弹板：连接上左两个点，右下两个点

这样可以将球的运动轨迹通过某种转化成为一个图

首先，观察到：

- 1 如果不打碎任何一个反弹板，那么球一定会循环的运动，循环运动可以看作图中的一个环
- 2 打碎一个反弹板，相当于将两个循环的运动变成一个

迷宫

所以，将一组循环的运动视作一个点，如果两个点之间可以通过打碎一个反弹板变成一个循环，那么这两个点之间视作有一条边

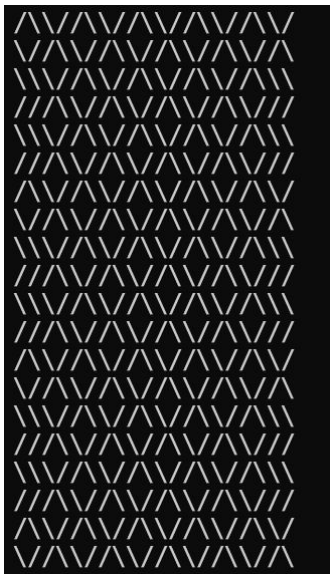
注意到询问的终点是没有方向的，所以正反的循环可以视作一个点

通过这种方式转化的图，一定是一个平面图

平面图上面，两个点之间的不超过两条边的最短路，可以通过类似 23 杭州 C 的方式完成

参考链接：https:

//qoj.ac/download.php?type=attachments&id=1516&r=1



魔法使考核

可以注意到翻倍操作的区间性是没用的，代价最低的做法一定是合理安排加一和翻倍操作的顺序，使得翻倍操作都是全局翻倍

考虑只有加一操作和全局翻倍时：

- 如果翻倍的代价低于加一的代价，那么一定是从高到低枚举每一位，把这一位是 1 的数字都加一，全部加完之后再翻倍，这样重复操作
- 如果翻倍代价更高，那么有可能的代价更低的方案是，先把高若干位都只用加一操作加出来，然后再开始加一、翻倍、加一、翻倍的循环，直接枚举两种操作的分界位即可

时间复杂度： $O(n \log \max(a_i))$