Tutorial

Problem 1

Task1 $n \leq 5000$

暴力或者打表均可。

Task2 $n \leq 10^{12}$

当 n 为偶数时，设 $m = \frac{n}{2} - 1$

当 n 为奇数时，设 $m = \frac{n-1}{2}$

可以发现， $n \bmod i \leq m$ ，且当 $i \leq m$ 时，有 $n \bmod (n-i) = i$ 。于是可以得出 $n \bmod i$ 取到 $[0, m]$ 的所有整数，因此答案会是 $2^k - 1$ ， k 的具体值判断一下即可

Problem 2

Task1 $n < 10^6$

考虑一个长度为i的序列的最后一个人，如果加入这个人，这个人要多久才能打到饭。

由此定义 $f[i][0/1]$ 表示最后一个人是 $0/1$ 的情况下，最后一个人打到饭的时间之和， 0 表示最后一个人是男生， 1 是女生。

定义 $g[i][0/1]$ 表示最后一个人是 $0/1$ 的情况下的总方案数。

$$
\begin{aligned}
f[i][0] &= f[i-a][0] + f[i-b][1] + d * g[i][0] \\
f[i][1] &= f[i-c][1] + f[i-b][0] + e * g[i][1] \\
g[i][0] &= g[i-a][0] + g[i-b][1] \\
g[i][1] &= g[i-c][1] + g[i-b][0]
\end{aligned}
$$

记得取模，复杂度 $O(n)$

Task2 $n < 10^{18}$

容易发现， n 非常大，但是 a, b, c 非常小。

因此每次转移时所需要的 $i-a, i-b, i-c$ 非常靠近 i ，因此可以考虑使用滚动数组转移。

但滚动数组并没有对时间上做出优化。

可以用矩阵乘法来代替滚动数组的转移，构造一个 $4 \cdot \max\{a, b, c\}$ 阶的转移矩阵即可。

复杂度 $O((4 \cdot \max\{a, b, c\})^3 \log n)$

Problem 3

设每个水晶球有一个对应的 1×4 矩阵 $\begin{pmatrix} A_i & B_i & C_i & 1 \end{pmatrix}$ ，对每个操作，可以得出对应的操作转移矩阵矩阵：

1. 火元素激发水元素能量：令 $A_i = A_i + B_i$ 。 $\begin{pmatrix}1&0&0&0\\1&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}$
2. 土元素激发火元素能量：令 $B_i = B_i + C_i$ 。 $\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&1&1&0\\0&0&0&1\end{pmatrix}$
3. 水元素激发土元素能量：令 $C_i = C_i + A_i$ 。 $\begin{pmatrix}1&0&1&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}$
4. 火元素能量定值增强：令 $A_i = A_i + v$ 。 $\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\v&0&0&1\end{pmatrix}$
5. 水元素能量翻倍增强：令 $B_i = B_i \times v$ 。 $\begin{pmatrix}1&0&0&0\\0&v&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}$
6. 土元素能量吸收融合：令 $C_i = v$ 。 $\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&0\\0&0&v&1\end{pmatrix}$

使用线段树维护区间内转移矩阵的和以及区间乘操作即可。

尽管开了原题的 2 倍时限，但还是要注意常数。

Problem 4

Task 1 $1 \le n,m \le 3$

枚举最后几位的情况，或者枚举三个字符串当前已经匹配了多少位，可以得出状态转移的形式，高斯消元即可

Task 2 $1 \le n,m \le 18$

建`AC`自动机跑高斯消元，做法与`[JSOI2009] 有趣的游戏`相同，时间复杂度 $O(n^3m^3)$ 。

Task 3 $n=2$

如果设 $f[i][j]$ 表示第一个串匹配了 i 位，第二个串匹配了 j 位的出现概率，那么状态数为 m^2 ，直接高斯消元是 $O(m^6)$ 的，或许有针对稀疏矩阵的高斯消元优化。

设两个人的串分别是 A,B ，假设当前游戏还没结束，且字符串为 S ，我们知道如果直接在 S 后面加上一个 A 游戏一定能结束，但是可能会出现在中途结束游戏的情况。

例：`A=101,B=110`

- 如果当前 S 以 `10` 结尾，那么加入第一个 `1` 的时候就直接结束，第一个人获胜，多加了 `01`
- 如果当前 S 以 `1` 结尾，那么加入 `10` 时结束，第二个人获胜，多加了 `1`
- 其余情况，第一个人获胜

于是可以得到一个等式： $S101 = A01 + B1 + A$ ，于是就有 $S \cdot \frac{1}{8} = A \cdot \frac{1}{4} + B \cdot \frac{1}{2} + A$ 。

这个式子的来源是，每个形如 $S101$ 的字符串都可以不重不漏地分解成三种不同的类型，于是这些字符串出现的概率必然相同。

我们再来看看直接在 S 后面加上一个 B 的情况，同样能够得到 $S110 = A10 + B$ ，于是就有 $S \cdot \frac{1}{8} = A \cdot \frac{1}{4} + B$ 。

通过上述两个式子我们是能够知晓 A, B 之间的关系的，但是并不方便我们直接得出结果。

我们还知道，一定有 $P_A + P_B = 1$ ，所以能够得到一个三元一次方程组 $\begin{cases} A + B = 1 \\ S = 10A + 4B \\ S = 2A + 8B \end{cases}$ ，解得 $\begin{cases} S = 6 \\ A = \frac{1}{3} \\ B = \frac{2}{3} \end{cases}$

于是在这种情况下，我们就得出了两人获胜的概率。

总结一下不难得出，当我们考虑加 A 串的时候，能够对式子产生贡献的情况是 A 串的一个前缀和 x 串（可能是 A 或 B ）的一个后缀发生了重合，假设重合部分的长度为 L ，那么式子右边就会多上 $x \cdot \frac{1}{2^{m-L}}$ 。

于是我们就可以得出一个标准化的式子：
$S \cdot \frac{1}{2^m} = \sum_{pre(A,L)=suf(A,L)} A \cdot \frac{1}{2^{m-L}} + \sum_{pre(A,L)=suf(B,L)} B \cdot \frac{1}{2^{m-L}}$

对加 B 串的情况，也能得到类似的结果，于是我们通过枚举 L 判断对应前后缀是否相同，并手工计算这两个方程里 A 和 B 的系数，就能算出 AB 之间的比值，从而求解。

Task 4 $1 \le n, m \le 300$

实际上**Task 3**的做法离正解已经无限接近了。

设 $pre(i, L)$ 表示第 i 个字符串长度为 L 的前缀， $suf(i, L)$ 同理，那么根据**Task 3**的结论，可以得出方程组的第 i 个式子为： $S \cdot \frac{1}{2^m} = \sum_{j=1}^{n} \sum_{pre(i,L)=suf(j,L)} P_j \cdot \frac{1}{2^{m-L}}$ ，再结合式子 $\sum_{i=1}^{n} P_i = 1$ ，即可联立方程组高斯消元求解。

接下去问题就在于如何快速判定 $pre(i, L) = suf(j, L)$ ，用各类字符串科技或者直接 `hash` 即可。