解题报告

杭州第二中学　叶卓睿

1 Tarot Sham Boast

1.1 题目大意

给你 s 个长度相同的字符集为 3 的字符串，你需要对它们按照在长为 n 的字符集为 3 的随机字符串中出现的概率排序，并输出这个顺序。

1.2 数据范围

- $1 \leq n \leq 10^6$
- $1 \leq s \leq 10$

1.3 解题过程

这是一道结论题，官方题解也只提供了一个非常长而不直观的证明。如果读者有更好的证明，可以联系这题的出题人，也欢迎来告知笔者。当然如果你对证明不感兴趣，可以直接跳到题解的最后。

1.3.1 准备工作

先明确一些定义：

- Δ 表示差分算子，即 $\Delta f(n) = f(n) - f(n-1)$。
- 记 P 为长度为 n 的随机串，$P_{l..r}$ 表示其下标在 $[l, r]$ 内的子串。
- 对于一个长度为 l 的模式串 s，定义集合 $S = \{1 \leq i < l \mid s_{1,l-i} = s_{i+1,l}\}$。
- 称模式串 s 在 i 处匹配，表示 $s = P_{i,i+l-1}$。

对于一个模式串，记 $f(n)$ 表示它在长度为 $n+l-1$ 的随机字符串中出现的概率，初值为 $\forall n \leq 0, f(n) = 0$，这是这道题目所关心的函数。

记 $g(n)$ 表示这个串只在 1 处出现，而没有在 $[2, n]$ 中匹配的概率。

则我们有

$$g(n) = \frac{1}{3^l} - \frac{\sum_{i \leq n-l} g(i)}{3^l} - \sum_{i \in S} \frac{g(n-i)}{3^i}$$

这个式子的意义是先算在 1 处出现的概率，然后枚举一个位置 $j+1$，答案减掉最后一次出现的位置为 $j+1$ 的概率，而这个概率是

$$\begin{cases} \frac{g(n-j)}{3^j} & j < l, j \in S \\ \frac{g(n-j)}{3^l} & j \geq l \\ 0 & \text{otherwise} \end{cases}$$

由于 $g(n) = \Delta f(n)$，带入 $g(n)$ 的递推式，可以得到

$$f(n) = \frac{n - \sum_{i \leq n-l} f(i)}{3^l} - \sum_{i \in S} \frac{f(n-i)}{3^i}$$

1.3.2 一些观察

引理 1： $f(n)$ 是单调函数，即对于所有 n，满足 $\Delta f(n) \geq 0$。

证明. 显然 $g(n) \geq 0$，因此 $\Delta f(n) = g(n) \geq 0$ □

引理 2： $\Delta f(n) \geq \frac{2}{3} \Delta f(n-1)$。

证明. 命题等价于 $g(n) \geq \frac{2}{3} g(n-1)$，这个可以用组合意义证明。对于一组 $n-1$ 的方案，即在 1 处匹配而没有在 $[2, n-1]$ 处匹配，那么加在随机串末尾的位置只要不是 s_l，则必然满足在 n 处没有匹配。也即 $g(n) \geq \frac{2}{3} g(n-1)$。 □

引理 3： 若集合 S 中包含了 i，则所有不超过 $l-1$ 的 i 的倍数都在 S 中。从而如果 $S \neq [1, l-1]$ 则 $1 \notin S$。

证明. 考虑字符串周期的性质，若 $\forall p, s_p = s_{p+i}$，则对于 $k \in N_+$ 必有若 $\forall p, s_p = s_{p+ik}$。 □

1.3.3 最终结论

对于 S, T 为两个不同模式串对应集合，不妨设 $r = \min T < \min S$。

我们的结论是 $f_S(n) \geq f_T(n)$，取等条件为 $n \leq r$。

证明. 记 $D(n) = f_S(n) - f_T(n)$，由 f 的定义我们得到

$$D(n) = -\frac{\sum_{i \leq n-l} D(i)}{3^l} - \sum_{i \in S} \frac{f_S(n-i)}{3^i} + \sum_{i \in T} \frac{f_T(n-i)}{3^i}$$

由此不难看出，$\forall n \le r, D(n) = 0$，以及 $D(r+1) = \frac{1}{3^{l+r}}$。

接下来我们试图归纳证明 $D(n) \ge \frac{2}{3}D(n-1)$，若此结论成立，则 $\forall n \ge 1, D(n) \ge 0$ 自动成立。

设结论在不超过 $n-1$ 时成立（归纳的基础一直到 $r+1$）。

则

$$\begin{aligned}\Delta D(n) &= -\frac{D(n-l)}{3^l} - \sum_{i\in S}\frac{\Delta f_S(n-i)}{3^i} + \sum_{i\in T}\frac{\Delta f_T(n-i)}{3^i} \\ &= -\frac{D(n-l)}{3^l} - \sum_{i\in S}\frac{\Delta D(n-i)}{3^i} - \sum_{i\in S\setminus T}\frac{\Delta f_T(n-i)}{3^i} + \sum_{i\in T\setminus S}\frac{\Delta f_T(n-i)}{3^i} \quad (1)\end{aligned}$$

由**引理 1**，以及 $r \in T\setminus S$，我们有

$$\sum_{i\in T\setminus S}\frac{\Delta f_T(n-i)}{3^i} \ge \frac{\Delta f_T(n-r)}{3^r} \tag{2}$$

连续应用**引理 2** 中的不等式，可以得到 $\Delta f_T(n-i) \le (\frac{3}{2})^{i-r}\Delta f_T(n-r)$，从而有

$$\begin{aligned}-\sum_{i\in S\setminus T}\frac{\Delta f_T(n-i)}{3^i} &\ge -\sum_{i>r}\frac{\Delta f_T(n-i)}{3^i} \\ &\ge \sum_{i>r}\frac{-(\frac{3}{2})^{i-r}\Delta f_T(n-r)}{3^i} \\ &= \sum_{i>r}-(\frac{1}{2})^{i-r}\frac{\Delta f_T(n-r)}{3^r} \\ &\ge -\frac{\Delta f_T(n-r)}{3^r} \quad (3)\end{aligned}$$

将(2)、(3)带入(1)式中，可得

$$\Delta D(n) \ge -\frac{D(n-l)}{3^l} - \sum_{i\in S}\frac{\Delta D(n-i)}{3^i}$$

由归纳假设，$D(n-i-1) \ge 0$，故 $\Delta D(n-i) \le D(n-i)$。此外由归纳假设 $D(n-i) \le (\frac{3}{2})^{i-1}D(n-1)$，进一步放缩可得

$$
\begin{aligned}
\Delta D(n) &\geq -\frac{D(n-l)}{3^l} - \sum_{i\in S}\frac{D(n-i)}{3^i} \\
&= -\sum_{i\in S\cup\{l\}}\frac{D(n-i)}{3^i} \\
&\geq -\sum_{i\geq \min S}\frac{D(n-i)}{3^i} \\
&\geq -\sum_{i\geq 2}\frac{D(n-i)}{3^i} \\
&\geq -\sum_{i\geq 2}\frac{D(n-1)}{3}(\frac{1}{2})^{i-1} \\
&\geq -\frac{D(n-1)}{3}
\end{aligned}
$$

因此，

$$
D(n) = D(n-1) + \Delta D(n) \geq \frac{2}{3}D(n-1)
$$

从而结论对 n 同样成立。 $\square$

这样一来，算法就很简单了，要比较两个串的概率，只需要取出所有满足 $2\cdot l - x \leq n$ 的 border x，比较从大到小排列后的 x 序列字典序，则字典序小的概率更大（或者说，比较集合 S 按从小到大排列，并在尾部加上一项 $+\infty$ 得到的序列的字典序，字典序大的概率更大）。集合 S 可以用 KMP 算法在线性时间内求出。

时间复杂度 $\mathcal{O}(ls\log s)$

2 Donut Drone

2.1 题目大意

有一个在两维上都首尾环绕在一起的 $n \times m$ 的矩形网格，每个格子的后继定义为右边、右上方、右下方这三个格子中权值最大的一个。

你会从 $(1,1)$ 的位置开始移动，移动操作为走到后继格子。共有 Q 次操作，操作共两种：

- 进行移动 k 次移动操作
- 修改某个格子的权值

2.2 数据范围

- $3 \le n, m \le 2000$
- $q \le 5000$
- 保证任何时候每个格子的后继唯一

2.3 解题过程

考虑维护第一列每个点跳一圈后回到第一列的什么位置，不妨记作 $f(i)$，那么对于询问点不在第一列的操作，只需要先暴力跳到第一列即可。

修改的话只需要找出所有第一列的点中有哪些位置最终能跳到当前修改点左侧、左上、左下中的一个，可以从当前列向前递推出受到影响的位置，不过过程中会涉及到 nm 个格子，因此还需要分析一些性质。

引理：往前递推的过程中受到影响的位置始终是个区间。

证明. 现在从第 $i+1$ 列推第 i 列，第 $i+1$ 列上的区间为 $[l,r]$，只需讨论区间非空的情况。

- 若 $l=r$，则新的位置集合是 $\{l-1,l,l+1\}$ 的子集。反证法，若新集合不是区间，则必为 $\{l-1,l+1\}$，由于 $l-1,l+1$ 的后继为 l，则 $a_{l,i+1} > a_{l-1,i+1}, a_{l,i+1} > a_{l-1,i-1}$。而这与 l 的后继不是 l 矛盾。

- 若 $l<r$，则新的位置集合 S 必然包含 $[l+1,r-1]$，待定位置为 $l-1,l,r,r+1$。反证法，若新集合不是区间，不妨设 $l-1\in S,l\notin S$，则 $l-1,l,l+1$ 这三个格子可以应用情况一的分析推出矛盾。另一种情况，$r+1\in S,r\notin S$ 也同理可以推出矛盾。

当然，如果过程中出现空集，则接下来一直都会是空集。 □

观察上述分析过程，可知每次递推都可以 $\mathcal{O}(1)$ 完成。所以只需向前递推这个区间，最后把得到的整个区间的 f 值修改为从这个特殊位置（询问点左侧、左上、左下）出发走到的第一列的位置。这里也可以使用 LCT 维护网格的后继，但是复杂度和常数较劣，正常的实现无法通过此题。

最后需要在第一列上求 k 级祖先，显然可以倍增维护，不过会比较慢。可以发现第一列的点由于出度为 1，因此是个内向基环树森林，由基环树性质，任何点的 n 级祖先一定在环上，因此可以先跳 n 步然后暴力求出周期（环长），简单模拟即可。

递推区间可能跨过 n ，为了减少细节，一种实现方式是认为 r 可以大于 n，相当于破环为链。

另外，递推区间的时候，别忘了特判 $l=r$ 以及空区间的情况。

时间复杂度 $O(nm+qm+qn)$。

3 Jump

3.1 题目大意

这是一道交互题。

有一个长度为偶数 n 的 01 串 S。你每次询问一个长度为 n 的 01 串 Q，交互库会计算 S,Q 中不同的位数，若这个值不是 $\frac{n}{2},n$ 中的一个，则返回 0，否则返回真实值。

你需要在 $n+500$ 次询问内问出 S。

3.2 数据范围

- $n \le 1000$
- n 是偶数

3.3 解题过程

首先，任何询问如果碰巧问出 n 则问题解决，接下来我们忽略这种情况。

显然只有询问出非零值才能获得有用信息，不妨先随机若干个 01 串，直至问出 $\frac{n}{2}$。这里分析一下期望询问次数。

对于一次询问，成功的概率为 $\binom{n}{\frac{n}{2}}2^{-n}$。

使用 Stirling 公式 $n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$ 对组合数进行估计，概率约为 $\sqrt{\frac{2}{\pi n}}$。

对于极限情况，$500 = \frac{n}{2}$，则连续失败 500 次的概率为

$$
\begin{aligned}
\left(1-\sqrt{\frac{2}{\pi n}}\right)^{\frac{n}{2}} &= \left(1-\sqrt{\frac{2}{\pi n}}\right)^{\sqrt{\frac{\pi n}{2}}\cdot\sqrt{\frac{n}{2\pi}}} \\
&\approx e^{-\sqrt{\frac{n}{2\pi}}} \\
&\approx 3\cdot 10^{-6}
\end{aligned}
$$

因此，我们几乎可以保证可以在 500 次以内问出一个 $\frac{n}{2}$ 的串 Q。

令 $a_i = Q_i \oplus S_i$，一个观察是，如果将 Q 的第 1、i 位反转后询问的结果仍是 $\frac{n}{2}$，就意味着 $a_1 \ne a_i$，否则意味着 $a_1 = a_i$。

因此，对于所有 $1 < i \leq n$，我们进行一次上述的询问，那么如果 a_1 确定，整个序列 a 就确定了，从而 $S = a \oplus Q$ 也确定下来了。

a_1 只有两种取值，分别询问一下则必然可以问出 S，问题就解决了。

总询问次数不超过 $n + 1 + \delta$，其中 δ 为第一部分的随机次数。

参考文献

[1] Wikipedia, Stirling's approximation

[2] WF2017 官方题解中提供的 K 题证明 `http://www.csc.kth.se/~austrin/icpc/tarotshamproof.pdf`