

Berlekamp-Massey 算法

算法介绍

设域 F 上有一无限数列 $s_0, s_1, \dots$ 我们想要对某个有限的 n 找到一个尽可能短的数列 $c_1, \dots, c_L$ 使得 $s_j = -\sum_{i=1}^L c_i s_{j-i}$ 对 $j=L, L+1, \dots, n-1$ 成立。特别地，若 $L=0$ 意味着 $s_0 = s_1 = \dots = s_{n-1} = 0$ 记 $s_0, \dots, s_{n-1}$ 为 $s^{(n)}$ 这样的数列 c 称为 $s^{(n)}$ 的递推式。由定义可以看出，任意 $L \leq n$ 的数列 c 都是 $s^{(n)}$ 的递推式。我们定义长度最小的递推式为最小递推式。下面证明一个重要的引理：

引理 1

设长度为 L 的数列 c 是 $s^{(n)}$ 的递推式，而不是 $s^{(n+1)}$ 的递推式；长度为 L' 的数列 c' 是 $s^{(n+1)}$ 的递推式，那么 $L' \geq n+1-L$

证明：采用反证法，假设 $L' \leq n-L$ 那么有
$$s_n = -\sum_{i=1}^{L'} c'_i s_{n-i} = -\sum_{i=1}^{L'} c'_i \left(-\sum_{j=1}^L c_j s_{n-i-j} \right)$$
 注意这里如果 $L' > n-L$ 则不能展开成括号内的形式
$$s_n = -\sum_{j=1}^{L'} c'_j \left(-\sum_{i=1}^{L'} c'_i s_{n-i-j} \right) = -\sum_{j=1}^{L'} c'_j s_{n-j}$$
 这与 c 不是 $s^{(n+1)}$ 的递推式矛盾 $\square$

我们定义 $L^{(n)}(s)$ 表示 $s^{(n)}$ 最小递推式的长度。显然有 $L^{(n)}(s) \leq L^{(n+1)}(s)$ 假如 $L^{(n)}(s)$ 对应的 c 不是 $s^{(n+1)}$ 的递推式，那么就有 $L^{(n+1)}(s) \geq \max\{L^{(n)}(s), n+1-L^{(n)}(s)\}$ 我们定义多项式 $C(x) = 1 + \sum_{i=1}^L c_i x^i$ 并记 $L^{(n)}(s)$ 对应的 c 为 $c^{(n)}$ 它对应的多项式为 $C^{(n)}(x)$

为了便于叙述下面的定理，我们先讲一个性质：如果不同的 $c^{(n)}$ 中，既有是 $s^{(n+1)}$ 的递推式的，也有不是 $s^{(n+1)}$ 的递推式的，那么显然有 $L^{(n)}(s) \geq n+1-L^{(n)}(s)$

定理 1

对于 $\forall n \in \mathbb{N}$ 若 $c^{(n)}$ 中有是 $s^{(n+1)}$ 的递推式的，那么 $L^{(n+1)}(s) = L^{(n)}(s)$ 若 $c^{(n)}$ 中有不是 $s^{(n+1)}$ 的递推式的，那么 $L^{(n+1)}(s) = \max\{L^{(n)}(s), n+1-L^{(n)}(s)\}$ 根据上面的性质，如果二者都有，命题也不矛盾。

证明：我们用归纳法证明该性质。记 $d_n = s_n + \sum_{i=1}^{L^{(n)}(s)} c^{(n)}_i s_{n-i}$

第一个部分，即存在一个 $d_n = 0$ 我们只要让 $c^{(n+1)} = c^{(n)}$ 即可。

第二个部分，即存在一个 $d_n \neq 0$ 若 $L^{(n)}(s) = 0$ 那么也显然成立。

否则，必然存在一个 m 使得 $L^{(m)}(s) < L^{(m+1)}(s) = \dots = L^{(n)}(s)$ 因为 $L^{(0)}(s) = 0$ 根据归纳假设，此时必然有 $L^{(n)}(s) = L^{(m+1)}(s) = m+1-L^{(m)}(s)$

定义多项式 $C^{(n+1)}(x) = C^{(n)}(x) - d_m^{-1} d_n x^{n-m} C^{(m)}(x)$ 计算可知 $C^{(n+1)}(x)$ 的次数就是 $\max\{L^{(n)}(s), n+1-L^{(n)}(s)\}$ 且常数项为 1。下面我们来验

证 $c^{(n+1)}$ 是 $s^{(n+1)}$ 的递推式。

记 $L = \max\{L^{(n)}(s), n+1-L^{(n)}(s)\}$ 对于 $i=L, L+1, \dots, n-1$

$$s_i + \sum_{j=1}^L c^{(n+1)}_j s_{i-j} = s_i + \sum_{j=1}^{L^{(n)}(s)} c^{(n)}_j s_{i-j} - d_{i-n} \left(s_{i-(n-m)} - \sum_{j=1}^{L^{(m)}(s)} c^{(m)}_j s_{i-(n-m)-j} \right) = 0$$

对于 $i=n$

$$s_n + \sum_{i=1}^L c^{(n+1)}_i s_{n-i} = s_n + \sum_{i=1}^{L^{(n)}(s)} c^{(n)}_i s_{n-i} - d_n \left(s_m - \sum_{i=1}^{L^{(m)}(s)} c^{(m)}_i s_{m-i} \right) = d_n - d_m^{(-1)} d_n d_m = 0$$

下面介绍两个实现时的数值分析：

性质 1

若 $s_n = XA^n Y$ 其中 $X \in F^{1 \times k}, A \in F^{k \times k}, Y \in F^{k \times 1}$ 那么对 $\forall n \in \mathbb{N}, L^{(n)}(s) \leq k$

证明：设 $p(x)$ 是 A 的特征多项式（若 k 次项为 -1 ，需要取反），根据 [Cayley-Hamilton theorem](#) $p(A) = 0$ 设 $c_i = p_{k-i}, i=1, \dots, k$ 对 $\forall i \geq k$

$$s_i + \sum_{j=1}^k c_j s_{i-j} = XA^i Y + \sum_{j=1}^k p_{k-j} XA^{i-j} Y = XC(A)A^{i-k} Y = 0$$

性质 2

设有无限数列 s 及长度为 L 的数列 c 满足 $\forall i \geq L, c$ 是 $s^{(i)}$ 的递推式。若存在长度为 $L' \leq L$ 的数列 c' 满足 c' 是 $s^{(2L)}$ 的递推式，那么对于 $\forall i \geq L' c$ 是 $s^{(i)}$ 的递推式。

证明：假设存在 $i \geq 2L$ 使得 c' 是 $s^{(i)}$ 的递推式，而不是 $s^{(i+1)}$ 的递推式，那么 $L+L' \geq i+1 \geq 2L+1$ 矛盾。

例题

Array Challenge

题源 [hdu 6172](#)

就是纯 BM

The number of circuits

题源：2018年牛客多校第9场 D

题目大意：给定 $k \leq 7$ 和 $n(2k+1 \leq n \leq 10^9)$ n 个点的有向图中边为 $i \rightarrow (i+j) \pmod n (\forall j \in [1, k])$ 问图中本质不同欧拉路的条数。

题解 [best theorem](#) 可知，答案为：

$$\det T_w(G) = \prod_{i=1}^n (\deg(i) - 1)! = \det T_w(G) \cdot ((k-1)!)^n$$

考虑计算 $\det T_w(G)$ 用 matrix-tree 定理，可以发现对于固定的 k, n 是满足线性递推关系的，我们用 BM 计算即可。

Expected Value

题源 Petrozavodsk Winter-2019. 300iq Contest 1 E

题目大意：给你一个连通平面图，开始在 1。每次等概率随机走到一个邻点，问第一次走到 n 的期望次数。

题解：由于可以用矩阵递推，考虑用 BM 求解。由于平面图满足 $E \leq 3V - 6$ 因此这部分的复杂度为 $\mathcal{O}(V^2)$

假设 $P(x) = \sum_{i=0}^{+\infty} p_i x^i$ 为递推式，设 $R(x) = P(x)Q(x)$ 显然 $R(x)$ 只有前 V 项不为 0。我们求即为
$$P'(1) = \frac{R'(1)}{Q'(1)} = \frac{R'(1)Q(1) - R(1)Q'(1)}{Q^2(1)}$$
 这部分的时间复杂度也为 $\mathcal{O}(V^2)$ 用 FFT 可以加速到 $\mathcal{O}(V \log V)$

时间复杂度 $\mathcal{O}(V^2)$

Fresh Matrix

题源 Petrozavodsk Winter-2018. ITMO U 1 Contest F

题目大意：定义一个 01 矩阵是好的，当且仅当所有 1 都不相邻，所有 0 相互四连通。问有多少种 $n \times m$ 的好的 01 矩阵，对质数取模 $n \leq 11, m \leq 10^9$

题解：对列记录该列有哪些格子是 1 ，以及 0 的格子之间的连通情况，状态有 2^{16} 个，转移有 96219 个。然后 BM 即可。

复杂度 $\mathcal{O}(ST + S^2 \log m)$ 如果你想用 FFT 优化到 $\mathcal{O}(ST + S \log S \log m)$ 那自然也是极好的。不过抄板可能会累死你：P

参考文献

[1] Massey J. Shift-register synthesis and BCH decoding[J]. IEEE transactions on Information Theory, 1969, 15(1): 122-127.

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
<https://wiki.cvbbacm.com/doku.php?id=technique:bm>

Last update: 2020/06/02 21:39

