这里是知识点部分的模板页面，请参照编写，将各部分修改为自己的内容

算法名称

算法简介（可选）

本页面给出了知识点 wiki 的基本格式。

算法内容 1

以下三级标题仅供参考。

性质 1

性质内容。

证明（可选）□

$$ 1+1=2\Box $$

算法内容 2

算法内容 3

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 $L'\ge n+1-L$□

证明：采用反证法，假设 $L'\le n-L$□那么有 $$ \begin{aligned} s_{n}&=-\sum_{i=1}^{L'}c'_{i}s_{n-i}\\ &=-\sum_{i=1}^{L'}c'_{i}\left(-\sum_{j=1}^{L}c_{j}s_{n-i-j}\right) \end{aligned} $$ 注意这里如果 $L'>n-L$□则不能展开成括号内的形式□ $$ \begin{aligned} s_{n}&=-\sum_{j=1}^{L}c_{j}\left(-\sum_{i=1}^{L'}c'_{i}s_{n-i-j}\right)\\ &=-\sum_{j=1}^{L}c_{j}s_{n-j} \end{aligned} $$ 这与 c 不是 $s^{(n+1)}$ 的递推式矛盾□$\Box$

我们定义 $L^{(n)}(s)$ 表示 $s^{(n)}$ 最小递推式的长度。显然有 $L^{(n)}(s)\le L^{(n+1)}(s)$□假如 $L^{(n)}(s)$ 对应的 c 不是 $s^{(n+1)}$ 的递推式，那么就有 $L^{(n+1)}(s)\ge\max\{L^{(n)}(s),n+1-L^{(n)}(s)\}$□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 $L^{(n)}(s)\ge n+1-L^{(n)}(s)$□

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 $L^{(m)}(s)<L^{(m+1)}(s)=\cdots=L^{(n)}(s)$□因为 $L^{(0)}(s)=0$□□根据归纳假设，此时必然有 $L^{(n)}=L^{(m+1)}(s)=m+1-L^{m}(s)$□

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 $i=L,L+1,\cdots,n-1$ $$ \begin{aligned} &s_{i}+\sum_{j=1}^{L}c^{(n+1)}_{j}s_{i-j}\\ =&s_{i}+\sum_{j=1}^{L^{(n)}(s)}c^{(n)}_{j}s_{i-j}-d^{-1}_{m}d_{n}\left(s_{i-(n-m)}-\sum_{j=1}^{L^{(m)}(s)}c^{(m)}_{j}s_{i-(n-m)-j}\right)\\ =&0 \end{aligned} $$ 对于 $i=n$ $$ \begin{aligned} &s_{n}+\sum_{i=1}^{L}c^{(n+1)}_{i}s_{n-i}\\ =&s_{n}+\sum_{i=1}^{L^{(n)}(s)}c^{(n)}_{i}s_{n-i}-d^{-1}_{m}d_{n}\left(s_{m}-\sum_{i=1}^{L^{(m)}(s)}c^{(m)}_{i}s_{m-i}\right)\\ =&d_{n}-d_{m}^{-1}d_{n}d_{m}\\ =&0\Box \end{aligned} $$ 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 $\forall n\in\mathbb{N},L^{(n)}(s)\le k$□

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□$p(A)=\mathcal{O}$□设 $c_{i}=p_{k-i},i=1,\cdots,k$□对 $\forall i\ge k$ $$ \begin{aligned} &s_{i}+\sum_{j=1}^{k}c_{j}s_{i-j}\\ =&XA^{i}Y+\sum_{j=1}^{k}p_{k-j}XA^{i-j}Y\\ =&XC(A)A^{i-k}Y\\ =&0\Box \end{aligned} $$

性质 2

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

证明：假设存在 $i\ge2L$□使得 c' 是 $s^{(i)}$ 的递推式，而不是 $s^{(i+1)}$ 的递推式，那么 $L+L'\ge i+1\ge2L+1$□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 $i\rightarrow (i+j)\text{ mod } n(\forall j \in [1, k])$，问图中本质不同欧拉路的条数。

题解：best theorem 可知，答案为：

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

考虑计算 $t_w(G)$，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 $E\le3V-6(V\ge3)$，因此这部分的复杂度为 $\mathcal{O}(V^{2})$。

假设 $P(x)=\sum_{i=0}^{+\infty}p_{i}x^{i}$，$Q(x)$ 为递推式，设 $R(x)=P(x)Q(x)$，显然 $R(x)$ 只有前 V 项不为 0。我们所求即为 $$ \begin{aligned} &P'(1)\\ =&(\frac{R}{Q})'(1)\\ =&\frac{R'(1)Q(1)-R(1)Q'(1)}{Q^{2}(1)} \end{aligned} $$ 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（$n\le11,m\le10^{9}$）

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 2161 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.