Petya and Sequence 解题报告

吉大附中实验学校 王天懿

November 2, 2015

1 题目描述

给出一个长度为N的序列$A_{0..N-1}$，问是否存在一个长度为N的序列$B_{0..N-1}$满足：
-至少存在一个$i(0 \le i \le N-1)$满足$B_i \ne 0$
-对于任意$0 \le j \le N-1$满足：

$$\sum_{i=0}^{N-1} A_i * B_{(i+j) mod\ N} = 0$$

T组数据。

2 数据规模与约定

$1 \le T \le 100$
$1 \le N \le 3 * 10^4$
$-1000 \le A_i \le 1000$
所有数据中N的总和不会超过$1.5 * 10^5$

3 题目解法1

问题等价于给出一个循环矩阵$C = \{A_0, A_1, ..., A_N\}$，问这个矩阵是否满秩。
一个思路是，我们可以考虑直接计算出这个矩阵的行列式，判断是否为0。

3.1 循环矩阵的行列式

现有一循环矩阵

$$A = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots & a_{n-1} \\ a_{n-1} & a_0 & a_1 & \cdots & a_{n-2} \\ a_{n-2} & a_{n-1} & a_0 & \cdots & a_{n-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & a_3 & \cdots & a_0 \end{bmatrix}$$

设函数$f(x) = a_0 + a_1 * x + a_2 * x^2 + ... + a_{n-1} * x^{n-1}$，那么有：

$$|A| = f(\epsilon_0) f(\epsilon_1) ... f(\epsilon_{n-1})$$

其中$\epsilon_k = w_n^k = e^{\frac{2\pi ki}{n}}$，即$\epsilon_0, \epsilon_1, ..., \epsilon_{n-1}$为方程$x^n = 1$在复数域$C$上的$n$个根。

3.1.1 循环矩阵行列式公式的证明

构造C上n阶方阵

$$B = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ \epsilon_0 & \epsilon_1 & \epsilon_2 & \cdots & \epsilon_{n-1} \\ \epsilon_0^2 & \epsilon_1^2 & \epsilon_2^2 & \cdots & \epsilon_{n-1}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \epsilon_0^{n-1} & \epsilon_1^{n-1} & \epsilon_2^{n-1} & \cdots & \epsilon_{n-1}^{n-1} \end{bmatrix}$$

那么由范德蒙矩阵[1]的相关知识可知，$|B| = \prod_{0 \le j < i \le n-1}(\epsilon_i - \epsilon_j)$，而$n$个单位根两两不同，故$|B| \neq 0$。
接下来考虑函数$f(x) = a_0 + a_1 * x + a_2 * x^2 + ... + a_{n-1} * x^{n-1}$，由于$\epsilon_k^n = 1$，我们有：
$f(\epsilon_k) = a_0 + a_1 * \epsilon_k + a_2 * \epsilon_k^2 + ... + a_{n-1} * \epsilon_k^{n-1}$
$\epsilon_k f(\epsilon_k) = a_{n-1} + a_0 * \epsilon_k + a_1 * \epsilon_k^2 + ... + a_{n-2} * \epsilon_k^{n-1}$
$\epsilon_k^2 f(\epsilon_k) = a_{n-2} + a_{n-1} * \epsilon_k + a_0 * \epsilon_k^2 + ... + a_{n-3} * \epsilon_k^{n-1}$
$\cdots$
$\epsilon_k^{n-1} f(\epsilon_k) = a_1 + a_2 * \epsilon_k + a_3 * \epsilon_k^2 + ... + a_0 * \epsilon_k^{n-1}$
故有

$$AB = \begin{bmatrix} a_0 & a_1 & \cdots & a_{n-1} \\ a_{n-1} & a_0 & \cdots & a_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ a_1 & a_2 & \cdots & a_0 \end{bmatrix} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \epsilon_0 & \epsilon_1 & \cdots & \epsilon_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \epsilon_0^{n-1} & \epsilon_1^{n-1} & \cdots & \epsilon_{n-1}^{n-1} \end{bmatrix}$$

$$= \begin{bmatrix} f(\epsilon_0) & f(\epsilon_1) & \cdots & f(\epsilon_{n-1}) \\ \epsilon_0 f(\epsilon_0) & \epsilon_1 f(\epsilon_1) & \cdots & \epsilon_{n-1} f(\epsilon_{n-1}) \\ \vdots & \vdots & \ddots & \vdots \\ \epsilon_0^{n-1} f(\epsilon_0) & \epsilon_1^{n-1} f(\epsilon_1) & \cdots & \epsilon_{n-1}^{n-1} f(\epsilon_{n-1}) \end{bmatrix}$$

注意到得到的矩阵中第i列有公因子$f(\epsilon_{i-1})$，可以提出行列式外。
故有：

$$|A| * |B| = |AB| = f(\epsilon_0) f(\epsilon_1) ... f(\epsilon_{n-1}) |B|$$

因为$|B| \neq 0$，故有：

$$|A| = f(\epsilon_0) f(\epsilon_1) ... f(\epsilon_{n-1})$$

3.2 计算循环矩阵的行列式

现在的问题是给出了公式，如何计算出行列式的值？
考虑快速傅里叶变换。
构造函数$F(x) = \sum_{i=0}^{n-1} a_i * w_{2n}^{i*i}, G(x) = \sum_{i=-n+1}^{n-1} w_{2n}^{-i*i}$
一个小技巧是FFT的时候如果$i + j \geq 2^k$的时候会把答案加在$(i + j) mod\ 2^k$次项上，因此只需要把$-i$次项放在$2^k - i$项的位置上即可。
考虑用FFT求出两个函数的积之后，得到的函数第$i(0 \le i \le n)$次项上会是什么？

$$[i](F(x) * G(x)) = \sum_{j=0}^{n-1} a_j * w_{2n}^{j*j} * w_{2n}^{-(i-j)*(i-j)} = \sum_{j=0}^{n-1} a_j * w_{2n}^{2ij - i*i}$$

而答案是什么？

$$|A| = \prod_{i=0}^{n-1} \sum_{j=0}^{n-1} a_j * w_n^{ij}$$

[1] http://en.wikipedia.org/wiki/Vandermonde_matrix

因此只需要对得到函数的第i项乘上一个w_{2n}^{i*i}即可。
为了避免精度问题，我们可以把这个问题转化到模意义上做。
寻找一个质数p满足$p = 2kn + 1$，然后找到p的一个原根g，用g^k来代替$2n$次单位根。
利用模任意素数的快速数论变换[2]代替快速傅里叶变换。
在p足够大的情况下(10^8级别？)如果计算出的行列式 $mod\ p = 0$，那么我们可以认为行列式为0。

4 时空复杂度

时间复杂度：$O(n \log n)$
空间复杂度：$O(n)$

5 题目解法2

抛开行列式的思路，我们考虑从秩上下手。
根据wiki[3]上的介绍，循环矩阵的秩满足以下公式：

$$rank(A) = n - degree(gcd(f(x), x^n - 1))$$

其中gcd代表两个多项式的最大公因式，$degree$代表最高项次数。
从这个式子中可以看出，如果$f(x)$与$x^n - 1$存在度数不为0的公因式，那么这个矩阵就是不满秩的。
利用FFT直接计算这两个多项式的最大公因式是$O(n^2 \log n)$的，我们要想其他办法优化。
这里我们介绍另一个本题相关的知识，叫做分圆多项式[4]。

5.1 分圆多项式

定义：$\varphi(n)$次整系数多项式$\phi_n(x)$满足$\phi_n(x) = \prod_{0 \le i < n, (i,n)=1} (x - w_n^i)$，其中$\phi_n(x)$称作分圆多项式。
例：
$\phi_1(x) = x - 1$
$\phi_2(x) = x + 1$
$\phi_3(x) = (x - w_3^1)(x - w_3^2) = x^2 + x + 1$
$\phi_4(x) = (x - w_4^1)(x - w_4^3) = x^2 + 1$
$\phi_5(x) = (x - w_5^1)(x - w_5^2)(x - w_5^3)(x - w_5^4) = x^4 + x^3 + x^2 + x + 1$
$\phi_6(x) = (x - w_6^1)(x - w_6^5) = x^2 - x + 1$
性质：
性质1：$\phi_i(x)$两两不同
性质2：$\phi_i(x)$不可约
性质3：$x^n - 1 = \prod_{d|n} \phi_d(x)$

回到我们的原问题。
由于$\phi_i(x)$不可约，因此我们可以将$x^n - 1 = \prod_{d|n} \phi_d(x)$看做$x^n - 1$的质因式分解，问题转化成了判断是否存在$\phi_d(x)$满足$d|n$且$\phi_d(x)|f(x)$。
直接做这个问题是非常困难的，但是我们可以转化一下。
引理1：$\phi_d(x)|f(x) \iff x^d - 1 | f(x) * \prod_{p\text{是质数}, p|d} (x^{\frac{d}{p}} - 1)$
想要证明这个引理，我们只需要知道以下事实：
$a|b \iff a * p_1 p_2 ... p_k | b * p_1^{a_1} p_2^{a_2} ... p_k^{a_k}$
其中$(a, p_1 p_2 ... p_k) = 1, a_i \ge 1 (1 \le i \le k)$
那么我们只需要枚举n的约数d，计算出$f(x) * \prod_{p\text{是质数}, p|d} (x^{\frac{d}{p}} - 1)$，判断是否是$x^d - 1$的倍数即可。
为了控制规模，我们还有以下引理：

[2]http://picks.logdown.com/posts/247168-fast-fourier-transform-modulo-prime
[3]http://en.wikipedia.org/wiki/Circulant
[4]http://en.wikipedia.org/wiki/Cyclotomic_polynomial

引理2：令$f(x)=\sum_{i\geq 0}a_ix^i$，那么$f(x)\ mod\ (x^n-1)=\sum_{i\geq 0}a_ix^{i\ mod\ n}$
证明很简单，考虑a_m*x^m减掉$a_m*x^{m-n}*(x^n-1)$会得到a_m*x^{m-n}，即指数上减掉一个n，所以取模后将会是$a_m*x^{m\ mod\ n}$
根据这个引理我们可以把多项式的规模控制在$O(d)$级别，故一次乘法的复杂度是$O(d)$的，而d的质数个数不会超过5，可以视作一个常数，因此验证$\phi_d(x)|f(x)$的复杂度不会超过$O(n)$。

5.2 具体做法

由于上述内容牵涉到了过多证明，我们不妨将做法再整理一遍。
1.枚举n的因数d。
2.新建数组$b_{0..d-1}$，其中$b_i=\sum_{j\ mod\ d=i}a_j$。
3.枚举d的质因子p，令$b'_i=b_{(i-\frac{d}{p})mod\ d}-b_i$，然后令$b=b'$
4.操作结束后若对于任意$0\leq i<d$满足$b_i=0$，则输出YES，否则枚举下一个因数d。
5.若对于所有d均无法满足条件，输出NO。

6 时空复杂度

时间复杂度：$O(5*n*d(n))$
空间复杂度：$O(n)$
其中$d(n)$为n的约数个数，不会超过96。

7 两种做法的比较

第一种做法相对便于理解，但代码较长，实现比较繁琐；第二种做法代码简单易实现，但需要大量的数学证明作为辅助，在我看来这两种做法各有各的优缺点，都是值得一写的做法。