不一般的DFT

周雨扬

SXYZ

2020 年 8 月 1 日

Contents

前言

本文需要以下前置知识:

* 快速傅里叶变换以及其原理(FFT)

* 一般模数快速傅里叶变换(MTT)

下面会对这两个算法进行简单的介绍。

约束与限定

在本文中，我们采用符号$|f(x)|$表示多项式$f(x)$的次数。这里我们约定任意非零次多项式的最高项系数非0。

在本文中，我们定义多项式$f(x)$在p处的点值为将p代入$f(x)$中得到的结果。

在本文中，我们采用符号ω_n表示n次单位根，也就是$\cos\left(\frac{2\pi}{n}\right)+\sin\left(\frac{2\pi}{n}\right)i$。

离散傅里叶变换(DFT)

离散傅里叶变换本质是将$\omega_n^0, \omega_n^1, \cdots, \omega_n^{n-1}$依次代入多项式$f(x)$中得到的点值序列$a$。

在下文中，我们称参数n为模长。这里需要需要保证$n > |a|$

离散傅里叶变换(DFT)

离散傅里叶变换本质是将$\omega_n^0, \omega_n^1, \cdots, \omega_n^{n-1}$依次代入多项式$f(x)$中得到的点值序列$a$。

在下文中，我们称参数n为模长。这里需要需要保证$n > |a|$

在已知点值序列的情况下，只需要求出$g(x) = \sum_{i=0}^{n-1} a_i x^i$在$\omega_n^0, \omega_n^{-1}, \cdots, \omega_n^{-(n-1)}$的点值序列$b$，则$f(x) = \sum_{i=0}^{n-1} b_i x^i$。证明略。

离散傅里叶变换(DFT)

离散傅里叶变换本质是将$\omega_n^0, \omega_n^1, \cdots, \omega_n^{n-1}$依次代入多项式$f(x)$中得到的点值序列$a$。

在下文中，我们称参数n为模长。这里需要需要保证$n > |a|$

在已知点值序列的情况下，只需要求出$g(x) = \sum_{i=0}^{n-1} a_i x^i$在$\omega_n^0, \omega_n^{-1}, \cdots, \omega_n^{-(n-1)}$的点值序列$b$，则$f(x) = \sum_{i=0}^{n-1} b_i x^i$。证明略。

快速傅里叶是一个基于分治的对于离散傅里叶变换的优化。单次运行复杂度为$O(n \log n)$。

一般模数离散傅里叶变换(MTT)

假设我们要求出来$A(x) \times B(x)$之后对p取模的结果。

由于double的精度问题，直接运行产生的结果可以达到$p^2 \times |A(x)|$的级别，精度无法接受。

一般模数离散傅里叶变换(MTT)

假设我们要求出来$A(x)\times B(x)$之后对p取模的结果。

由于double的精度问题，直接运行产生的结果可以达到$p^2\times|A(x)|$的级别，精度无法接受。

此时考虑把多项式的系数表示成$xC+y$的形式，其中$0\leq y<c$。此时我们分别对于x,y运行DFT，对于$xC+y$的乘法直接暴力合并点值，最后计算答案。

取$C=\sqrt{p}$,此时产生的结果只有$p\times|A(x)|$的级别，可以接受。

一般模数离散傅里叶变换(MTT)

假设我们要求出来$A(x) \times B(x)$之后对p取模的结果。

由于double的精度问题，直接运行产生的结果可以达到$p^2 \times |A(x)|$的级别，精度无法接受。

此时考虑把多项式的系数表示成$xC + y$的形式，其中$0 \le y < c$。此时我们分别对于x, y运行DFT，对于$xC + y$的乘法直接暴力合并点值，最后计算答案。

取$C = \sqrt{p}$,此时产生的结果只有$p \times |A(x)|$的级别，可以接受。

在毛啸同学2016年的集训队论文中有提到对于该算法的优化，有兴趣的读者可以自行查阅。

北大集训D2T1

多次询问，每次询问$(x+1)^K$对(x^n-1)取模之后的$x^m(m<n)$次项系数，对质数p取膜。保证在模p意义下存在n次单位根。

$\sum n \le 10^6, p \le 10^7, 0 \le K \le 10^9$

算法

设ω表示模意义下的n次单位根。

算法

设ω表示模意义下的n次单位根。

$$
\begin{aligned}
\sum_{i=0}^{\infty}\binom{K}{in+m} &= \sum_{i=0}^{K}\binom{K}{i}\frac{\sum_{j=0}^{j<n}\omega^{(i-m)j}}{n} \\
&= \frac{1}{n}\sum_{j=0}^{j<n}\omega^{-mj}\sum_{i=0}^{K}\binom{K}{i}\omega^{ij} \\
&= \frac{1}{n}\sum_{j=0}^{j<n}\omega^{-mj}(1+\omega^{j})^{K}
\end{aligned}
$$

算法

设ω表示模意义下的n次单位根。

$$
\begin{aligned}
\sum_{i=0}^{\infty}\binom{K}{in+m} &= \sum_{i=0}^{K}\binom{K}{i}\frac{\sum_{j=0}^{j<n}\omega^{(i-m)j}}{n} \\
&= \frac{1}{n}\sum_{j=0}^{j<n}\omega^{-mj}\sum_{i=0}^{K}\binom{K}{i}\omega^{ij} \\
&= \frac{1}{n}\sum_{j=0}^{j<n}\omega^{-mj}(1+\omega^{j})^{K}
\end{aligned}
$$

直接暴力求和，复杂度$O(\sum n\log K)$

北大集训D2T1加强

多次询问，每次给定次数小于n 的多项式$A(x)$，询问$A(x)^K$对(x^n-1)取膜之后的多项式，系数质数p取膜。保证在模p意义下存在n 次单位根。

$\sum n \le 10^6, p \le 10^7, 0 \le K \le 10^9$

算法

直接推式子不好推。

暴力求power时间复杂度$O(n\log n\log K)$,会TLE。

由于FFT本质是循环卷积，我们可以进行模长为n的DFT,但是直接暴力是$O(n^2)$的。

算法

直接推式子不好推。

暴力求power时间复杂度$O(n\log n\log K)$,会TLE。

由于FFT本质是循环卷积，我们可以进行模长为n的DFT,但是直接暴力是$O(n^2)$的。

如果对于模长为n的DFT我们可以在$O(n\log n)$的复杂度内解决，则对于本问题可以实现$O(n(\log n+\log K))$的复杂度。

但是快速傅里叶变换仅能处理模长为2^n的特殊情况，因此我们需要更加好的处理办法。

基于分治算法的一般模长DFT

考虑去扩展DFT中的分治算法。

假设现在需要计算$V(j)=\sum\limits_{i=0}^{i<n} a_i\omega_n^{ij}$的值。

基于分治算法的一般模长DFT

考虑去扩展DFT中的分治算法。

假设现在需要计算$V(j)=\sum\limits_{i=0}^{i<n} a_i\omega_n^{ij}$的值。

枚举$V(j)$在模n最小质因子d意义下的值D。设$m=n/d$,做一下简单的推导可得

$$\begin{aligned}V(jd+D) &= \sum_{i=0}^{i<n} \omega_n^{i(jd+D)} a_i \\ &= \sum_{i=0}^{i<n} \omega_n^{ijd}\omega_n^{iD} a_i \\ &= \sum_{i=0}^{i<n} \omega_m^{ij}\omega_n^{iD} a_i\end{aligned}$$

基于分治算法的一般模长DFT

设$B(y)=\sum_{x=0}^{x<d}\omega_n^{(xm+y)D}a_{xm+y}$,则有

$$
\begin{aligned}
V(jd+D) &= \sum_{i=0}^{i<n}\omega_m^{ij}\omega_n^{iD}a_i \\
&= \sum_{y=0}^{y<m}\sum_{x=0}^{x<d}\omega_m^{xmj}\omega_m^{yj}\omega_n^{(xm+y)D}a_{xm+y} \\
&= \sum_{y=0}^{y<m}\sum_{x=0}^{x<d}\omega_m^{yj}\omega_n^{(xm+y)D}a_{xm+y} \\
&= \sum_{y=0}^{y<m}\omega_m^{yj}B(y)
\end{aligned}
$$

基于分治算法的一般模长DFT

在枚举完D之后，我们可以在$O(n)$的复杂度内计算出$B(i)$，同时将其转化为一个规模为l/d的子问题。这样子的操作总共会进行d轮。

基于分治算法的一般模长DFT

在枚举完D之后，我们可以在$O(n)$的复杂度内计算出$B(i)$，同时将其转化为一个规模为l/d的子问题。这样子的操作总共会进行d轮。

如果产生的子问题大小为1则可以直接计算，否则我们不断使用这种策略将其划分为若干个规模和不变的子问题。

基于分治算法的一般模长DFT

因为在分治过程中，分治树上每一层的大小总和恒定不变，为n。对于某一层，d轮操作会将n个元素整体扫d次，因此总时间复杂度为$O(n * \sum d)$。

基于分治算法的一般模长DFT

因为在分治过程中，分治树上每一层的大小总和恒定不变，为n。对于某一层，d轮操作会将n个元素整体扫d次，因此总时间复杂度为$O(n * \sum d)$。

但是由于$\sum d$规模仍然为$O(n)$级别，我们需要一个复杂度更加优秀的算法。

基于卷积的一般模长DFT

我们再次回到DFT的式子上来。

$$V(j) = \sum_{i=0}^{i<n} \omega_n^{ij} a_i$$

基于卷积的一般模长DFT

我们再次回到DFT的式子上来。

$$V(j) = \sum_{i=0}^{i<n} \omega_n^{ij} a_i$$

考虑ij的实际含义，可以看成有两堆物品，第一堆大小为i，第二堆大小为j，从每堆中选出一个物品的方案数。

基于卷积的一般模长DFT

我们再次回到DFT的式子上来。

$$V(j)=\sum_{i=0}^{i<n}\omega_n^{ij}a_i$$

考虑ij的实际含义，可以看成有两堆物品，第一堆大小为i，第二堆大小为j，从每堆中选出一个物品的方案数。

我们在这个方案数上考虑进行容斥，现在我们将两堆物品合并并在其中选择两个互异元素，减去在同时第一堆中选两个和同时第二堆中选两个的方案，即为从每堆中选出一个物品的方案数。

转化成数学公式即为:$\binom{i+j}{2}-\binom{i}{2}-\binom{j}{2}=ij$。

基于卷积的一般模长DFT

将上式代入DFT式，得到

$$V(j)\omega_n^{\binom{j}{2}} = \sum_{i=0}^{i<n} a_i \omega_n^{-\binom{i}{2}} \omega_n^{\binom{i+j}{2}}$$

基于卷积的一般模长DFT

将上式代入DFT式，得到

$$V(j)\omega_n^{\binom{j}{2}} = \sum_{i=0}^{i<n} a_i \omega_n^{-\binom{i}{2}} \omega_n^{\binom{i+j}{2}}$$

显然这个式子可以被看成是$C(x)=\omega_n^{-\binom{x}{2}}$对$B(x)=\sum a_x\omega_n^{\binom{x}{2}}$做的一次减法卷积。

基于卷积的一般模长DFT

将上式代入DFT式，得到

$$V(j)\omega_n^{\binom{j}{2}} = \sum_{i=0}^{i<n} a_i \omega_n^{-\binom{i}{2}} \omega_n^{\binom{i+j}{2}}$$

显然这个式子可以被看成是$C(x) = \omega_n^{-\binom{x}{2}}$对$B(x) = \sum a_x \omega_n^{\binom{x}{2}}$做的一次减法卷积。

此时我们只关心卷积结果，而不关心卷积的模长问题，因此可以直接接MTT求解。

时间复杂度$O(n \log n)$

基于卷积的一般模长DFT

将上式代入DFT式，得到

$$V(j)\omega_n^{\binom{j}{2}} = \sum_{i=0}^{i<n} a_i \omega_n^{-\binom{i}{2}} \omega_n^{\binom{i+j}{2}}$$

显然这个式子可以被看成是$C(x) = \omega_n^{-\binom{x}{2}}$对$B(x) = \sum a_x \omega_n^{\binom{x}{2}}$做的一次减法卷积。

此时我们只关心卷积结果，而不关心卷积的模长问题，因此可以直接MTT求解。

时间复杂度$O(n \log n)$

小优化技巧：原本这里的MTT模长为$3*n$，但是由于我们不关心前面n个元素的正确性。因此可以利用FFT循环卷积的特性，将模长开到$2*n$级别即可。

基于卷积的一般模长IDFT

类似的，对于IDFT式，得到

$$V(j)\omega_n^{-\binom{j}{2}} = \sum_{i=0}^{i<n} a_i \omega_n^{\binom{i}{2}} \omega_n^{-\binom{i+j}{2}}$$

基于卷积的一般模长IDFT

类似的，对于IDFT式，得到

$$V(j)\omega_n^{-\binom{j}{2}} = \sum_{i=0}^{i<n} a_i \omega_n^{\binom{i}{2}} \omega_n^{-\binom{i+j}{2}}$$

显然这个式子仍可以被看成是一次减法卷积，直接MTT求解。

时间复杂度$O(n \log n)$

基于卷积的一般模长IDFT

类似的，对于IDFT式，得到

$$V(j)\omega_n^{-\binom{j}{2}} = \sum_{i=0}^{i<n} a_i \omega_n^{\binom{i}{2}} \omega_n^{-\binom{i+j}{2}}$$

显然这个式子仍可以被看成是一次减法卷积，直接MTT求解。
时间复杂度$O(n \log n)$
这种算法也被称为`bluestein` 算法或者`Z` 变换。

UOJ 500 任意基DFT

有n次多项式$f(x)=\sum\limits_{i=0}^{i\le n} a_ix^i$。

Q次询问，第i次询问$f(q_i)$对998244353取膜的结果

q_i按照如下方式生成:

$\forall 1\le i\le Q, q_i=(q_{i-1}\times qx+qy)\ mod\ 998244353$

$1\le n\le 2.5\times 10^5, 1\le Q\le 10^6$,$2\le qx<998244353$,$0\le q_0,qy<998244353$

UOJ 500 任意基DFT

假设我们已知n次多项式$f(x)$,则我们可以:

UOJ 500 任意基DFT

假设我们已知n次多项式$f(x)$,则我们可以:

$O(n)$找到一个n次多项式$g(x)$ 满足$g(x)=f(x*k)$。

UOJ 500 任意基DFT

假设我们已知n次多项式$f(x)$,则我们可以:

$O(n)$找到一个n次多项式$g(x)$ 满足$g(x) = f(x * k)$。

$O(n \log n)$找到一个n 次多项式$h(x)$ 满足$h(x) = f(x + k)$。

UOJ 500 任意基DFT

假设我们已知n次多项式$f(x)$,则我们可以:

$O(n)$找到一个n次多项式$g(x)$ 满足$g(x) = f(x * k)$。

$O(n \log n)$找到一个n 次多项式$h(x)$ 满足$h(x) = f(x + k)$。

同时观察到询问的值比较特殊，我们尝试用过若干次上述变换将其转化成比较优美的形式。

UOJ 500 任意基DFT

不难发现$q_i = q_0 q x^i + \sum_{j=0}^{j<i} q y q x^j$。

UOJ 500 任意基DFT

不难发现$q_i = q_0qx^i + \sum_{j=0}^{j<i} qyqx^j$。

将其乘以$qx-1$，得到$q_i = q_0(qx-1)qx^i + qyqx^i - qy$

UOJ 500 任意基DFT

不难发现$q_i = q_0 qx^i + \sum\limits_{j=0}^{j<i} qyqx^j$。

将其乘以$qx-1$，得到$q_i = q_0(qx-1)qx^i + qyqx^i - qy$

将其加上qy，得到$q_i = q_0(qx-1)qx^i + qyqx^i$

UOJ 500 任意基DFT

不难发现$q_i = q_0 qx^i + \sum_{j=0}^{j<i} qyqx^j$。

将其乘以$qx-1$，得到$q_i = q_0(qx-1)qx^i + qyqx^i - qy$

将其加上qy，得到$q_i = q_0(qx-1)qx^i + qyqx^i$

将其除以$q_0(qx-1)+qy$，得到$q_i = qx^i$

注意在$q_0(qx-1)+qy$为0时，询问值均为q_0，此处需要特殊判断。

UOJ 500 任意基DFT

同时我们对$f(x)$做同样的变换，即求出n 次多项式$g(x)$满足$g(x) = f(\frac{x(q_0(qx-1)+qy)-qy}{qx-1})$。

由于在题目给定条件下所有变换步骤均合法,所以有且仅有一个满足条件的n次多项式$g(x)$。

UOJ 500 任意基DFT

同时我们对$f(x)$做同样的变换，即求出n 次多项式$g(x)$满足$g(x)=f(\frac{x(q_0(qx-1)+qy)-qy}{qx-1})$。

由于在题目给定条件下所有变换步骤均合法,所以有且仅有一个满足条件的n次多项式$g(x)$。

现在我们需要求出多项式$g(x)$在$qx^1,qx^2,\cdots,qx^n$处的点值。即求出$V(i)=\sum\limits_{j=0}^{j\le n}a_jqx^{ij}$。

类比于一般模长DFT，我们仍然可以将上面的式子改写成一次减法卷积的形式。

UOJ 500 任意基DFT

同时我们对$f(x)$做同样的变换，即求出n 次多项式$g(x)$满足$g(x)=f(\frac{x(q_0(qx-1)+qy)-qy}{qx-1})$。

由于在题目给定条件下所有变换步骤均合法,所以有且仅有一个满足条件的n次多项式$g(x)$。

现在我们需要求出多项式$g(x)$在$qx^1,qx^2,\cdots,qx^n$处的点值。即求出$V(i)=\sum_{j=0}^{j\le n}a_jqx^{ij}$。

类比于一般模长DFT，我们仍然可以将上面的式子改写成一次减法卷积的形式。

时间复杂度$O(n\log n+Q\log Q)$。

UOJ 498 新年的追逐

定义两个简单无向图$G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ 的乘积为一个新的图$G_1 \times G_2 = (V^*, E^*)$。其中新的点集$V^* = \{(a,b)|a \in V_1, b \in V_2\}$，新的边集$E^* = \{((u_1, v_1), (u_2, v_2)) \mid (u_1, u_2) \in E_1, (v_1, v_2) \in E_2\}$。

UOJ 498 新年的追逐

定义两个简单无向图$G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ 的乘积为一个新的图$G_1 \times G_2 = (V^*, E^*)$。其中新的点集$V^* = \{(a, b) | a \in V_1, b \in V_2\}$, 新的边集$E^* = \{((u_1, v_1), (u_2, v_2)) \mid (u_1, u_2) \in E_1, (v_1, v_2) \in E_2\}$。

给定正整数n，以及n个正整数$m_1, m_2, \cdots, m_n$。你需要求出新图$H = (((G_1 \times G_2) \times G_3) \times \cdots) \times G_n$的期望连通块数量对998244353取模的结果。其中G_i所有包含m_i个节点的图中等概率随机生成的。

$n, m_i \leq 100000$

UOJ 498 新年的追逐

我们考虑给我们一组G_k 的序列之后如何计算连通块数量。

UOJ 498 新年的追逐

我们考虑给我们一组G_k 的序列之后如何计算连通块数量。

首先考虑在每个图中选一个点，如果某个选的点数的度数为0（我们称其为“孤立点”），则这个点序列在H 上对应的点是孤立点。

否则我们考虑每个图中选择一个大小>1 的连通块。考虑这些连通块的乘积得到的连通块有多少。注意到现在每个点都有邻边，且是无向图，因此我们可以在一条边上反复走。两个点序列之间的可达性可以简化为路径长度的奇偶性。

UOJ 498 新年的追逐

我们考虑给我们一组G_k 的序列之后如何计算连通块数量。

首先考虑在每个图中选一个点，如果某个选的点数的度数为0（我们称其为“孤立点”），则这个点序列在H 上对应的点是孤立点。

否则我们考虑每个图中选择一个大小> 1 的连通块。考虑这些连通块的乘积得到的连通块有多少。注意到现在每个点都有邻边，且是无向图，因此我们可以在一条边上反复走。两个点序列之间的可达性可以简化为路径长度的奇偶性。

如果两个点在一个存在奇环的图中，那么显然奇数长度和偶数长度的路径都有。

如果两个点在一个二分图中，那么这和他们是否在同一部中有关。

因此我们可以得到：如果选的这n 个连通块中有k 个不存在奇环，那么这些连通块的乘积将会给答案贡献$2^{\max(k-1,0)}$ 个连通块。

UOJ 498 新年的追逐

因此我们只需要知道全体大小为m_k 的图可以有多少个孤立点，多少个无奇环的连通块，多少个连通块，则可以由此算出答案。

UOJ 498 新年的追逐

因此我们只需要知道全体大小为m_k 的图可以有多少个孤立点，多少个无奇环的连通块，多少个连通块，则可以由此算出答案。

我们考虑染色二分图的EGF：$B = \sum_{n\geq 0}\sum_{m\geq 0}\frac{\binom{n+m}{n}2^{nm}x^{n+m}}{(n+m)!}$，则无奇环的连通块显然恰有2 种方法染色，可以得到EGF 为$\frac{\ln B}{2}$，无奇环连通块数量可以通过$\frac{G\ln B}{2}$ 表示。

UOJ 498 新年的追逐

因此我们只需要知道全体大小为m_k 的图可以有多少个孤立点，多少个无奇环的连通块，多少个连通块，则可以由此算出答案。

我们考虑染色二分图的EGF：$B=\sum_{n\ge0}\sum_{m\ge0}\frac{\binom{n+m}{n}2^{nm}x^{n+m}}{(n+m)!}$，则无奇环的连通块显然恰有2 种方法染色，可以得到EGF 为$\frac{\ln B}{2}$，无奇环连通块数量可以通过$\frac{G\ln B}{2}$ 表示。

这里我们可以采用bluestain 算法来优化求染色二分图的EGF 的过程。

时间复杂度$O(n\log n)$。

CODECHEF POLYEVAL

给定n次多项式$f(x) = \sum_{i=0}^{i \le n} a_i x^i$。

Q次询问$f(y)$对$786433(3 * 2^{18} + 1)$取模的值。

$n, Q \le 250000$。

算法1

这是一道多项式多点求值好题!

时间复杂度$O(n\log^2 n)$。

算法1

这是一道多项式多点求值好题!

时间复杂度$O(n\log^2 n)$。

直接拉出来板子交了上去。

可惜T了。

算法2

考虑DFT的实际含义，不难发现模长为n 的DFT可以看成是将$\omega_n^0, \omega_n^1, \cdots, \omega_n^{n-1}$ 代入多项式$A(x)$产生的点值。

算法2

考虑DFT的实际含义，不难发现模长为n 的DFT可以看成是将$\omega_n^0,\omega_n^1,\cdots,\omega_n^{n-1}$ 代入多项式$A(x)$产生的点值。

由于mo存在原根，这说明存在g满足$\omega_{p-1}\equiv g \pmod p$。同时根据原根的定义，$g^0,g^1,\cdots,g^{p-2}$ 在模意义下互不相同且恰能取到所有与p互质的小于p的正整数。

算法2

考虑DFT的实际含义，不难发现模长为n 的DFT可以看成是将$\omega_n^0, \omega_n^1, \cdots, \omega_n^{n-1}$ 代入多项式$A(x)$产生的点值。

由于mo存在原根，这说明存在g满足$\omega_{p-1} \equiv g \pmod p$。同时根据原根的定义，$g^0, g^1, \cdots, g^{p-2}$ 在模意义下互不相同且恰能取到所有与p互质的小于p的正整数。

这说明如果对$A(x)$跑一轮模长为$mo-1$的DFT，我们可以得到$1, 2, \cdots, mo-1$的点值。同时，对于0处的点值，显然其为a_0.

时间复杂度$O(p \log p)$

推广

类似的，由于p^c存在原根，这说明存在g满足$\omega_{\phi(p^c)} \equiv g \pmod{p^c}$。同时根据原根的定义，$g^0, g^1, \cdots, g^{\phi(p^c)-1}$ 在模意义下互不相同且恰能取到所有与p^c互质的小于p^c的正整数。这一部分我们仍然可以使用一般模长DFT解决。

推广

类似的，由于p^c存在原根，这说明存在g满足$\omega_{\phi(p^c)} \equiv g (\bmod\ p^c)$。同时根据原根的定义，$g^0, g^1, \cdots, g^{\phi(p^c)-1}$ 在模意义下互不相同且恰能取到所有与p^c互质的小于p^c的正整数。这一部分我们仍然可以使用一般模长DFT解决。

对于与p^c不互质的整数，设其为x，显然$x^c \equiv 0 (\bmod\ p^c)$,这说明我们仅需要考虑小于$c$次的项即可。

推广

类似的，由于p^c存在原根，这说明存在g满足$\omega_{\phi(p^c)} \equiv g (\bmod\ p^c)$。同时根据原根的定义，$g^0, g^1, \cdots, g^{\phi(p^c)-1}$ 在模意义下互不相同且恰能取到所有与p^c互质的小于p^c的正整数。这一部分我们仍然可以使用一般模长DFT解决。

对于与p^c不互质的整数，设其为x，显然$x^c \equiv 0 (\bmod\ p^c)$,这说明我们仅需要考虑小于$c$次的项即可。

由于$c = O(\log mo)$，总复杂度不变，仍为$O(mo \log mo)$

推广

特殊的，对于$2^c(c \geq 3)$的特殊模数，虽然其不存在原根，但是可以证明$\pm 3^0, \pm 3^1, \cdots, \pm 3^{2^{c-2}}$在模意义下互不相同且均与2互质。证明部分较为复杂此处略去。

推广

特殊的，对于$2^c(c \geq 3)$的特殊模数，虽然其不存在原根，但是可以证明$\pm 3^0, \pm 3^1, \cdots, \pm 3^{2^{c-2}}$在模意义下互不相同且均与2互质。证明部分较为复杂此处略去。

此时我们仍然可以使用bluestein算法分两步解决这一部分的问题。

推广

特殊的，对于$2^c(c \geq 3)$的特殊模数，虽然其不存在原根，但是可以证明$\pm 3^0, \pm 3^1, \cdots, \pm 3^{2^{c-2}}$在模意义下互不相同且均与2互质。证明部分较为复杂此处略去。

此时我们仍然可以使用bluestein算法分两步解决这一部分的问题。

类似的，与2不互质的仍然只需要考虑小于c次的项，因此总复杂度仍为$O(mo \log mo)$。

推广

特殊的，对于$2^c(c \geq 3)$的特殊模数，虽然其不存在原根，但是可以证明$\pm 3^0, \pm 3^1, \cdots, \pm 3^{2^{c-2}}$在模意义下互不相同且均与2互质。证明部分较为复杂此处略去。

此时我们仍然可以使用bluestein算法分两步解决这一部分的问题。

类似的，与2不互质的仍然只需要考虑小于c次的项，因此总复杂度仍为$O(mo \log mo)$。

若使用中国剩余定理合并不同p^c的答案，每一个p^c使用上述算法解决，则对于任意模数均可做到$O(mo \log mo)$的多项式多点求值。

Thanks

感谢人大附中邓明扬同学为本文审稿。

谢谢大家，预祝在冬令营取得好成绩!