【知识总结】快速傅里叶变换（FFT）

这可能是我第五次学 FFT 了…… 菜哭 qwq

先给出一些个人认为非常优秀的参考资料：

一小时学会快速傅里叶变换（Fast Fourier Transform） - 知乎

小学生都能看懂的 FFT！！！ - 胡小兔 - 博客园

快速傅里叶变换（FFT）用于计算两个n次多项式相乘，能把复杂度从朴素的$O(n^2)$优化到$O(nlog_2n)$。一个常见的应用是计算大整数相乘。

本文中所有多项式默认x为变量，其他字母均为常数。所有角均为弧度制。

一、多项式的两种表示方法

我们平时常用的表示方法称为“系数表示法”，即

$$A(x)=\sum_{i=0}^{n}a_ix^i$$

上面那个式子也可以看作一个以x为自变量的n次函数。用$n+1$个点可以确定一个n次函数（自行脑补初中学习的二次函数）。所以，给定$n+1$组x和对应的$A(x)$，就可以求出原多项式。用$n+1$个点表示一个n次多项式的方式称为“点值表示法”。

在“点值表示法”中，两个多项式相乘是$O(n)$的。因为对于同一个x，把它代入A和B求值的结果之积就是把它带入多项式$A\times B$求值的结果（这是多项式乘法的意义）。所以把点值表示法下的两个多项式的$n+1$个点的值相乘即可求出两多项式之积的点值表示。

~~线性复杂度点值表示好哇好~~

但是，把系数表示法转换成点值表示法需要对$n+1$个点求值，而每次求值是$O(n)$的，所以复杂度是$O(n^2)$。把点值表示法转换成系数表示法据说也是$O(n^2)$的（然而我只会$O(n^3)$的高斯消元 qwq）。所以暴力取点然后算还不如直接朴素算法相乘……

但是有一种神奇的算法，通过取一些具有特殊性质的点可以把复杂度降到$O(nlog_2n)$。

二、单位根

从现在开始，所有n都默认是2的非负整数次幂，多项式次数为$n-1$。应用时如果多项式次数不是2的非负整数次幂减1，可以加系数为0的项补齐。

先看一些预备知识：

复数$a+bi$可以看作平面直角坐标系上的点(a,b)。这个点到原点的距离称为模长，即$\sqrt{a^2+b^2}$；原点与(a,b)所连的直线与实轴正半轴的夹角称为辐角，即$sin^{-1}\frac{b}{a}$。复数相乘的法则：模长相乘，辐角相加。

把以原点为圆心，1为半径的圆（称为“单位圆”）n等分，n个点中辐角最小的等分点（不考虑1）称为n次单位根，记作

ω_n ，则这n个等分点可以表示为$\omega_n^k(0 \le k < n)$

这里如果不理解，可以考虑周角是2π ，n次单位根的辐角是$\frac{2\pi}{n}$ 。$w_n^k = w_n^{k-1} \times w_n^1$ ，复数相乘时模长均为1，相乘仍为1。辐角$\frac{2\pi(k-1)}{n}$ 加上单位根的辐角$\frac{2\pi}{n}$ 变成$\frac{2\pi k}{n}$ 。

单位根具有如下性质：

1. 折半引理

$$w_{2n}^{2k} = w_n^k$$

模长都是1，辐角$\frac{2\pi \times 2k}{2n} = \frac{2\pi k}{n}$ ，故相等。

2. 消去引理

$$w_n^{k+\frac{n}{2}} = -w_n^k$$

这个从几何意义上考虑，$w_n^{k+\frac{n}{2}}$ 的辐角刚好比w_n^k 多了$\frac{2\pi \times \frac{n}{2}}{n} = \pi$ ，刚好是一个平角，所以它们关于原点中心对称。互为相反数的复数关于原点中心对称。

3.（不知道叫什么的性质）其中k是整数

$$w_n^{a+kn} = w_n^a$$

这个也很好理解：w_n^n 的辐角是2π ，也就是转了一整圈回到了实轴正半轴上，这个复数就是实数1。乘上一个w_n^n 就相当于给辐角加了一个周角，不会改变位置。

三、离散傅里叶变换（DFT）

DFT 把多项式从系数表示法转换到点值表示法。

我们大力尝试把n次单位根的0到$n-1$ 次幂分别代入$n-1$ 次多项式$A(x)$ 。首先先对$A(x)$ 进行奇偶分组，得到：

$$A_1(x)=\sum_{i=0}^{\frac{n-1}{2}}a_{2i}\cdot x^i$$

$$A_2(x)=\sum_{i=0}^{\frac{n-1}{2}}a_{2i+1}\cdot x^i$$

则有：

$$A(x)=A_1(x^2)+x\cdot A_2(x^2)$$

把w_n^k 代入，得：

$$A(w_n^k)=A_1(w_n^{2k})+w_n^k\cdot A_2(w_n^{2k})$$

根据折半引理，有：

$$A(w_n^k)=A_1(w_{\frac{n}{2}}^k)+w_n^k\cdot A_2(w_{\frac{n}{2}}^k)$$

此时有一个特殊情况。当$\frac{n}{2}\leq k<n$ ，记$a=k-\frac{n}{2}$ ，则根据消去引理和上面第三个性质，有：

$$w_n^a=-w_n^k$$

$$w_{\frac{n}{2}}^a=w_{\frac{n}{2}}^k$$

所以

$$A(w_n^k)=A_1(w_{\frac{n}{2}}^a)-w_n^a\cdot A_2(w_{\frac{n}{2}}^a)$$

这样变换主要是为了防止右侧式子里出现w_n 的不同次幂。

按照这个式子可以递归计算。共递归$O(log_2n)$ 层，每层需要$O(n)$ 枚举k，因此可以在$O(nlog_2n)$ 内把系数表示法变为点值

表示法。

四、离散傅里叶反变换（IDFT）

设$w_n^k(0\le k<n)$ 代入多项式$A(x)$ 后得到的点值为b_k，令多项式$B(x)$ ：

$$B(x)=\sum_{i=0}^{n-1}b_ix^i$$

一个结论：设$w_n^{-k}(0\le k<n)$ 代入$B(x)$ 后得到的点值为c_k，则多项式$A(x)$ 的系数$a_k=\frac{c_k}{n}$ 。下面来证明这个结论。

$$\begin{aligned}c_k&=\sum_{i=0}^{n-1}b_i\cdot w_n^{-ik}\\&=\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}a_j\cdot w_n^{ij}\cdot w_n^{-ik}\\&=\sum_{j=0}^{n-1}a_j\sum_{i=0}^{n-1}w_n^{i(j-k)}\end{aligned}$$

脑补一下$\sum_{i=0}^{n-1}w_n^{i(j-k)}$ 怎么求。可以看出这是一个公比为w_n^{j-k} 的等比数列。

当$j=k$ ，$w_n^0=1$ ，所以上式的值是n。

否则，根据等比数列求和公式，上式等于$w_n^{j-k}\cdot\frac{w_n^{n(j-k)}-1}{w_n^{j-k}-1}$ 。$w_n^{n(j-k)}$ 相当于转了整整$(j-k)$ 圈，所以值为1，这个等比数列的和为0。

由于当$j\neq k$ 时上述等比数列值为0，所以$c_k=a_kn$ ，即$a_k=\frac{c_k}{n}$

至此，已经可以写出递归的 FFT 代码了。（~~常数大的一批 qwq~~

实测 洛谷 3803 有77 分，会 TLE 两个点。

五、优化

递归太慢了，我们用迭代。

考虑奇偶分组的过程。每一次把奇数项分到前面，偶数项分到后面，如$\{a_0,a_1,a_2,a_3,a_4,a_5,a_6,a_7\}$，按照这个过程分组，最终每组只剩一个数的时候是$\{a_0,a_4,a_2,a_6,a_1,a_5,a_3,a_7\}$。经过仔 mo 细 bai 观 da 察 lao，发现$1_{(10)}=001_{(2)}$，$4_{(10)}=100_{(2)}$，一个数最终变成的数的下标是它的下标的二进制表示颠倒过来（并不知道为什么）。我们可以递推算这个（其中 lg2 是$log_2 n$）：

```
rev[i] = rev[i >> 1] >> 1 | ((i & 1) << (lg2 - 1))
```

可以先生成原数组经过$log_2 n$次奇偶分组的最终状态，然后一层一层向上合并即可。

另外，标准库中的三角函数很慢，可以打出w_n^k和w_n^{-k}的表（或者只打一个表，因为$w_n^{-k}=w_n^{n-k}$）。当前分治的区间长度为l时，查询w_l^k相当于查询$w_n^{\frac{nk}{l}}$（这里要小心nk爆 int…… 血的教训）。

全文完