卷积的优化和应用

安师大附中 罗哲正

什么是卷积？

- 数学上卷积的定义是这样的：
- $(f * g)(\tau) \int_{-\infty}^{+\infty} f(x)g(\tau - x)$
- 当然在信息学竞赛中我们一般都使用离散的形式，即：
- $(f * g)(k) = \sum_{x=-\infty}^{+\infty} f(x)g(k - x)$
- 换一种写法就是$(f * g)_k = \sum f_i g_j [i + j = k]$
- 循环卷积：$(f * g)_k = \sum f_i g_j [i + j \equiv k (\mathrm{mod}\, n)]$
- 卷积意义的理解：多项式乘法

如何计算卷积$f * g = h$?

- 花式做法(si)：
- 枚举k，枚举i, j，若$i + j = k$，则$h_k \leftarrow h_k + f_i g_j$

- 枚举i, j，$h_{i+j} \leftarrow h_{i+j} + f_i g_j$即可。
- 时间复杂度$O(n^2)$。

- 可不可以快一点？

傅里叶变换

- 函数$f(x)$的傅里叶变换定义为：
- $F(\omega) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt$
- 可以导出逆离散傅里叶变换：
- $f\,(t) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} F(\omega)e^{i\omega t}d\omega$
- 卷积定理：函数卷积的傅里叶变换是函数傅里叶变换的乘积（一个域中的卷积相当于另一个域中的乘积）。
- $h = f * g \Leftrightarrow H(x) = F(x)G(x)$

离散傅里叶变换

- 序列f的离散傅里叶变换(DFT)定义为：
- $F_k = \sum_{j=0}^{n-1} f_j e^{\frac{-2\pi ijk}{n}}$
- 逆变换为：
- $f_j = \frac{1}{n}\sum_{k=0}^{n-1} F_k e^{\frac{2\pi ijk}{n}}$
- 卷积定理依旧适用：序列循环卷积的傅里叶变换是序列傅里叶变换的点积。
- $h = f * g \Leftrightarrow H_k = F_k G_k$

数论变换

- 在模意义下可以使用原根的方幂来代替单位根，具体令 $\omega_n = g^{\frac{P-1}{n}}$。
- 则数论变换(NTT)定义为：
- $F_k = \sum_{j=0}^{n-1} f_j {\omega_n}^{jk}$
- 逆变换为：
- $f_j = \frac{1}{n} \sum_{k=0}^{n-1} F_k {\omega_n}^{-jk}$
- 在模意义下卷积定义依然成立。

离散傅里叶变换的点值理解

- 令$\omega_n = e^{\frac{-2\pi i}{n}}$，考虑多项式$f(x) = \sum_{k=0}^{n-1} f_k x^k$
- $F_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} f_j e^{\frac{-2\pi ijk}{n}} = f\left(\omega_n{}^k\right)$
- 即将n个单位根带入多项式得到的点值。
- 那么IDFT的过程就相当于插值。

SRM592 1000pts SplittingFoxes2

- 有一个长度为n循环序列A，满足$A_i = A_{n-i}$，给出A的循环卷积，求字典序最小的A。
- $n \leq 25.$

- 循环多项式开方，考虑DFT的点值意义，先DFT，每一项开方，再IDFT，由于复数开方有两个根，需要2^n枚举所有情况，由对称性可以省去一半的枚举量，复杂度$O(2^n n^2)$。

某不知来源的题

- 给一个n次多项式P，给Q个x，求P在这些x处的点值，对$M = 786433(3 \times 2^{18} + 1, 一个质数)$取模。
- $n, Q \leq 250000.$

- 直接对多项式做大小为M的NTT即可，FFT是采用分治优化的，于是第一次分治采用基-3的分治。

快速傅里叶变换

- 快速傅里叶变换(FFT)是能在$O(n\log n)$时间内计算$n=2^k$的序列的DFT的算法。

- 已经非常普及了对吧，于是这里就不讲了= =

Codechef NOV 12 **COUNTARI**

- 给定一个长度为n序列A，求有多少长度为3的子序列为等差序列，两个子序列不同当且仅当有一项下标不同。
- $n \le 100000; 1 \le A_i \le 30000.$

- 分块，大小为 S ，三个至少有两个在同一块的情况可以暴力枚举，复杂度$O(nS)$，否则考虑中间一项在块内的情况，可以对两边的值域做卷积，复杂度$O\left(\frac{nA\log A}{S}\right)$，取$S=\sqrt{A\log A}$，复杂度就是$O\left(n\sqrt{A\log A}\right)$。

2016杭二春节集训 人种鉴定器

- 游戏中卡牌的稀有度有N, R, SR, UR四种，其中稀有度最高的UR卡画面精美，技能强大，Cenbo很想得到。花费虚拟货币Loveca可以进行一次抽卡，有1%的概率抽到UR，9%的概率抽到SR，90%的概率抽到R。此外还有保证能得到一张SR或UR的十一连抽卡，它相当于在进行十一次抽卡之后，如果十一张都是R则将最后一张以10%的概率变成UR，90%的概率变成SR。
  Cenbo进行了$N1$次普通抽卡和$N2$次十一连抽卡，但是只得到了M张UR。他想知道他的运气击败了百分之多少的玩家，也就是求进行同样的抽卡后得到UR的数量小于M的概率。
- $N1 \le 100000; N2 \le 10000.$

2016杭二春节集训 **人种鉴定器**

- 令A_i表示进行一次普通抽卡之后抽到i张UR的概率，B_i表示进行一次十一连抽之后抽到i张UR的概率。
- 将A和B看成生成函数，则答案的生成函数就是$A^{N1}B^{N2}$。
- 卷积采用FFT优化，倍增是$O(n\log^2 n)$的。
- 实际上直接把点值N次方即可，使用快速幂计算，复杂度$O(n\log n)$。

2016雅礼省选前集训 第K大C

- 给定二维数组$a_{0\cdots n-1,0\cdots m-1}$和$b_{0\cdots n-1}$定义$c_i = \sum_{j=0}^{n-1} a_{j,b_{i\times j \bmod n}}$，求$c$中的第$K$大数。
- $1 \le n \le 250000; 1 \le m \le 4; n$是质数.

- 枚举$k = b_{i\times j \bmod n}$，令$d_j = [b_j = k]$。
- 计算$c_i = \sum_{j=0}^{n-1} a_{j,k}\, d_{i\times j \bmod n}$并求和即可，特判掉下标为0的情况，找出$n$的原根$g$并对下标求离散对数可以转化成：
- $c_i = \sum_{j=0}^{n-2} a_{j,k}\, d_{i+j \bmod n-1}$将$d$翻转就变成了卷积的形式，使用FFT优化即可，复杂度$O(mn\log n)$。

2012集训队互测 binomial

- 对于给定的n和p，求对于所有的$0 \le i < p$, 满足$\binom{n}{k} \bmod p = i$的k的个数。
- $p = 51061; n < p^{10}$.

- 考虑Lucas定理$\binom{n}{m} \equiv \prod \binom{n_i}{m_i} \bmod p$，其中$0 \le n_i, m_i < p; n = \sum n_i p^i ; m = \sum m_i p^i$，显然需要$m_i \le n_i$才有意义。
- 直接设$f_{i,j}$表示考虑了不超过p^i的部分，模p为j的方案数。
- 转移时求出$b_{i,j}$表示$\binom{n_i}{m_i} = j$的方案数，则$f_{i,j} \to f_{i,jk \bmod p} b_{i,k}$
- 对下标求离散对数可以使用FFT优化，复杂度$O(p \log p \log n)$。

SRM 603 1000pts SumOfArrays

- 给定两个长度为n序列A_i, B_i，你需要把A, B重新排列，令$C_i = A_i + B_i$，你需要使得C中的众数出现次数最多。
- $n \le 100000.$

- 令a_i, b_j, c_k分别为A, B, C中i, j, k的出现次数
- 其实就是求$c_k = \sum \min(a_i, b_j)\,[i + j = k]$，枚举$\min(a_i, b_j)$，不超过10就FFT，超过10就暴力。

更为广泛的卷积定义

- 对于一种二元运算$\oplus$，卷积定义为
- $(f * g)_k = \sum f_i g_j [i \oplus j = k]$

- 显然当$\oplus$是$+$的时候，就是普通卷积的定义。
- 当$\oplus$同构于循环群的运算的时候，就是循环卷积的定义。

- 下面我们来研究一下$\oplus$是位运算的时候的卷积。

快速沃尔什变换

- 快速沃尔什变换(FWT)可以用来快速计算位运算卷积。
- 位运算每一维是独立的，所以只要对$(a_0, a_1) * (b_0, b_1)$构造。
- 对于XOR运算，有：
- $(A_0, A_1) = (a_0 + a_1, a_0 - a_1); (a_0, a_1) = \left(\frac{A_0+A_1}{2}, \frac{A_0-A_1}{2}\right)$
- 对于AND运算，有：
- $(A_0, A_1) = (a_0 + a_1, a_1); (a_0, a_1) = (A_0 - A_1, A_1)$
- 对于OR运算，有：
- $(A_0, A_1) = (a_0, a_1 + a_0); (a_0, a_1) = (A_0, A_1 - A_0)$

快速沃尔什变换

- 实际计算的时候，先对两边分别FWT，然后使用2项的FWT变换两边即可。
- 与DFT一样，FWT有如下性质。
- $a * b = c \Leftrightarrow A_k B_k = C_k$
- 于是我们就可以在$O(n\log n)$时间里愉快的计算位运算卷积啦。

CROC2016 Final C.Binary Table

- 给定一个$n \times m$的01矩阵，你可以取反若干行，再取反若干列，要使得矩阵中1的个数最少。
- $n \le 20; m \le 100000.$

- 每一列看成一个二进制数，设C_i表示状态为i的列的数目。
- 令B_j表示行确定之后列状态为j最少1数目，则
- $B_j = \min\{cnt(j), n - cnt(j)\}$
- 令行取反状态为k，则显然$i\ XOR\ k = j \Rightarrow i\ XOR\ j = k$
- A_k表示行状态为k的最小1个数，把B和C做XOR卷积即可得到A。

2016湖南省选前集训 兔子的晚会

- 有$2n+1$只兔子，刚开始第i只兔子的幸福指数是$a_i \in [0,m]$，若存在某个$x \in [L,R]$满足a_i+x的异或和是0，则称a_i是幸福的，求幸福的序列的方案数，对质数取模。
- $1 \le n,m,L,R \le 1000.$

- 若对于每个序列和x的组合求方案数，可以枚举x，然后令$f_i = \left[i \in [x,m+x]\right]$，把$f$做异或卷积$2n+1$次方即可。
- 从低位到高位可以推得对于一个序列a只有唯一一个x满足条件，于是这样计数就不重复了。

TCO 2012 2A EvenPaths

- 给定一个n点m边有向无环图，编号为$0\sim n-1$，有k个点可能是障碍点也可能不是障碍点，其他点一定不是障碍点，现在问你2^k种情况中有多少种情况使得从0号点到1号点恰有偶数条不经过障碍点的路径。
- $n \le 50; m \le 500; k \le 32.$

- 按拓扑序将k个点均分为两个集合A, B，考虑一种枚举，枚举A中集合哪些是障碍，求出从0到A中每个点的合法方案数，枚举B中哪些是障碍，求出从A中每个点不经过A中的点到达1的合法方案数，那么以路径在A中经过的最后一个点为关键点可以合并方案。

TCO 2012 2A EvenPaths

- 这个算法是$O(2^k)$的，考虑优化合并。
- 枚举A集合中哪些是障碍点，求出这种状态下0到A中每个点i的方案数的奇偶性，可以状态压缩成一个$\frac{k}{2}$位二进制数。
- 对B集合也一样枚举障碍点，求出A中每个点不经过A中其他点到达1的方案数奇偶性，依旧是压缩成一个$\frac{k}{2}$位二进制数。
- 考虑合并两个状态，枚举A中经过的最后一个点并合并其实相当于把二进制中的每一位做乘法，而二进制位的乘法相当于AND运算，于是可以采用FWT优化，复杂度$O\left(2^{\frac{k}{2}}(n+m)+k2^{\frac{k}{2}}\right)$。

FFT本身的应用

- FFT除了用来计算卷积之外，式子本身也有一些特别的应用。

- 下面看一道题感受一下……

CTSC2014 小秘密

- 有一个01串加密算法，确定K_1, K_2两个密钥，算法如下：
- $X_0 = K_1$
- 依次读入每一个位I_p，并对它进行一下计算
  - $T_p = w\left(f\left(X_p\right) \ AND \ K_2\right)$
  - $O_p = I_p \ XOR \ T_p$
  - $X_{p+1}\left(2X_p + O_p\right) \bmod 2^N$
- 输出O_p
- 其中$f(x) = (1994x^5 + 11x^4 + 4x^3 + 1995x^2 + 9x + 24) \bmod 2^N$
- $w(x)$表示x二进制中1的个数的是否为奇数。

CTSC2014 小秘密

- 现在已知明文是随机01串，出现0的概率是$\frac{1}{2}+b$，现在给你足够长的密文(设长度是L)，你需要求出K_2(一个64位二进制数)。
- $L=2^{23}; N \le 60; 0.12 \le b \le \frac{1}{2}$.

CTSC2014 小秘密

- 首先每一步的$f(X_p)$的值是可以直接算出来的接着考虑式子：
- $T_p = w(f(X_p) AND K_2)$
- $w(x)$可以看成模二意义的数位和，而AND运算可以看成按位乘法。
- 所以其实每一位就是一个方程，设$K_2 = \sum_{j=0}^{N-1} 2^j x_j, x_j \in \{0,1\}$。
- 则每一个方程的就是$\sum_{j=0}^{N-1} a_i x_i \equiv 0 \bmod 2$，其中有38%的噪声。
- 把变量分成三段，每段20位，按第二段分组并组内消元，再按第三段分组并组内消元，可以得到至少6×2^{20}个方程，考虑噪声影响，一个新的方程由4个原方程得到，噪声出现奇数次的概率是0.49834112。
- 于是就是解20元带噪声模2意义下线性方程组。

CTSC2014 小秘密

- $\sum_{j=0}^{n-1} a_j x_j \equiv 0 \bmod 2$可以转化为$\prod_{j=0}^{n-1}(-1)^{a_i x_j} = 1$。
- 考虑$n = 1$的情况，则左边答案的和关于K_2的关系可以写成：
- $(F_0, F_1) = ((-1)^{0\times 0} f_0 + (-1)^{0\times 1} f_1, (-1)^{1\times 0} f_0 + (-1)^{1\times 1} f_1)$
- 观察这个式子：其实就是大小为2的FFT。
- 多个变量其实就是多维的情况，于是可以使用一个20维的FFT来求出每一个位置的答案和。

FWT与FFT的关系

- 这里主要考虑XOR卷积，可以发现在$n = 2$的时候，异或卷积和循环卷积等价。
- 而当$n = 2^k$的时候，其实可以把序列分成两个2^{k-1}的序列，分别FWT之后，再对两个序列做一次$n = 2$的FFT。
- 于是$n = 2^k$的FWT其实就是每一维长度为2的k维FFT。

CTSC2014 小秘密

- 那么我们直接使用大小为2^{20}的FWT来计算20维FFT就好了。
- 接着我们轮换变量，就可以解出$x_{[20,39]}, x_{[40,59]}$。

更为广泛的位运算卷积

- 在上一题中，我们使用FWT解决了高维DFT问题，那么我们从DFT的角度来看FWT。
- XOR可以看成每一位模2的加法考虑推广XOR卷积，把下标看成一个向量$(i_0, i_1, \cdots, i_{k-1})$, 两个向量相加看成每一维在模意义下做加法即：
- $(i_0, i_1, \cdots, i_{k-1}) + (j_0, j_1, \cdots, j_{k-1}) = (l_0, l_1, \cdots, l_{k-1})$
- 其中$l_t = i_t + j_t \bmod p_t$
- 显然XOR卷积就是$p_t = 2$的情况。
- 对于p_t是任意整数的情况，做第t维大小为p_t的高维DFT即可。

更为广泛的位运算卷积

- XOR可以推广为每一位模2的加法即$l_t = i_t + j_t \bmod 2$
- AND和OR可以转化成min和max。
- $AND: l_t = \min\{i_t, j_t\}$ $OR: l_t = \max\{i_t, j_t\}$

- 这样的卷积能不能使用FWT优化呢？

继续推广FWT

- 普通位运算FWT的证明是构造性的，考虑换一个角度来观察，以OR为例，将下标的二进制数看成一个集合。
- FWT的过程其实就是求了子集和。
- IFWT的过程其实就是对子集和容斥得到原数组。

- 一个集合的子集任意去并集都还是这个子集的子集，所以OR卷积对子集和来说就是点积。

继续推广FWT

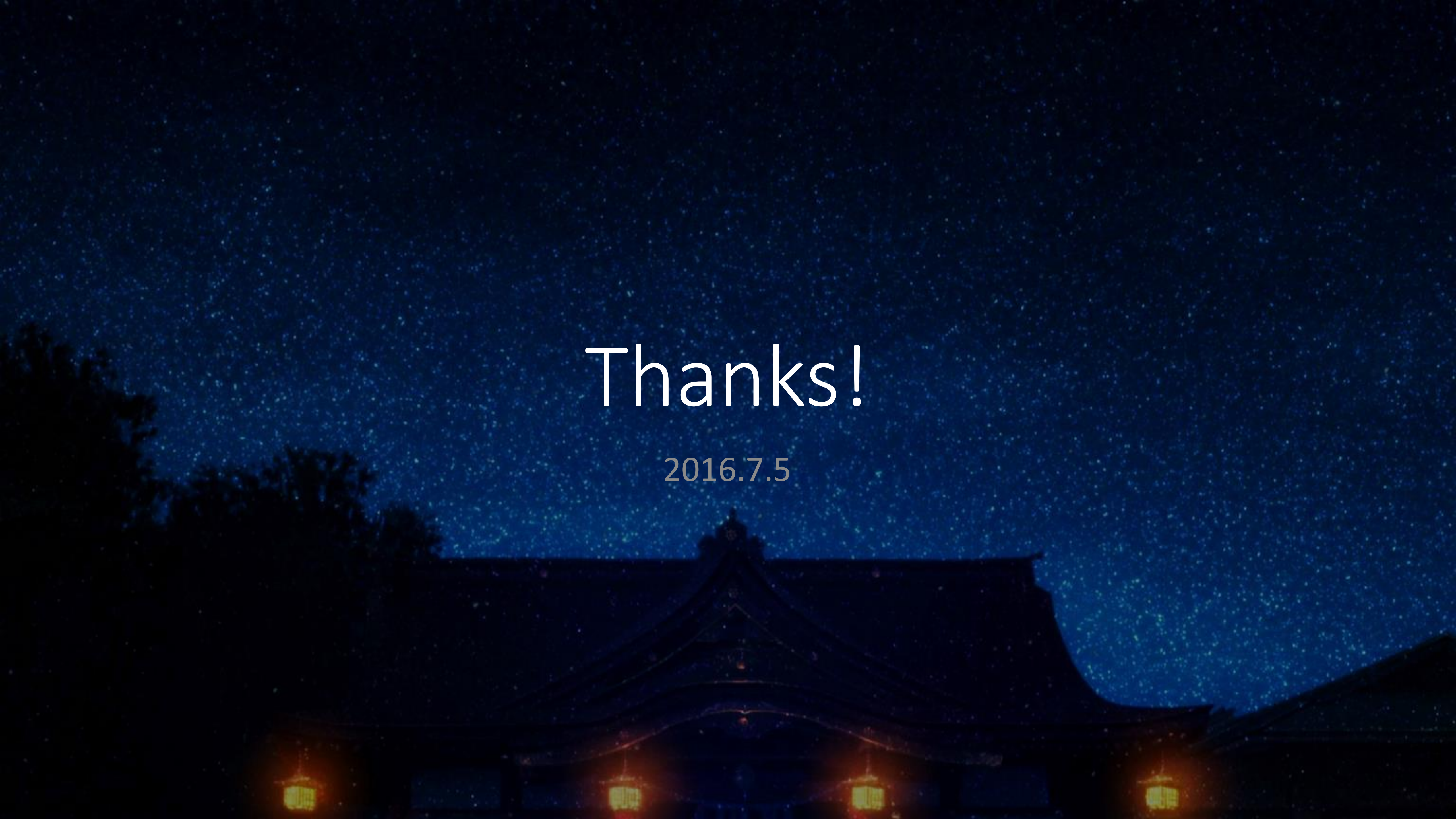
- 考虑这样的FWT：
- $F_{(i_0,i_1,\cdots,i_{k-1})} = \sum_{j_t \le i_t} f_{(j_0,j_1,\cdots,j_{k-1})}$
- 和对应的IFWT：
- $f_{(i_0,i_1,\cdots,i_{k-1})} = \sum_{i_t - 1 \le j_t \le i_t} (-1)^{\sum_{t=0}^{k-1} i_t - j_t} F_{(j_0,j_1,\cdots,j_{k-1})}$
- 其实FWT就是对每一维求前缀和，IFWT是对每一维差分。

SRM 691 900pts Undivisors

- 给一个$n \times m$的矩阵a，接下来进行如下操作
- 1随机一个子矩阵b
- 2随机一个子矩阵c
- 3令$S = b \cup c$
- 4找出最小的正整数x使得$x \nmid LCM(S)$
- 求x的期望。
- $1 \le n, m \le 50; 1 \le a_{i,j} \le 61.$

SRM 691 900pts Undivisors

- 考虑LCM，其实就是对每个质因子的次数求max。
- 于是直接把61以内每个质数的指数作为一维，做“OR”卷积即可。
- 可以使用FWT优化。


Thanks!
2016.7.5