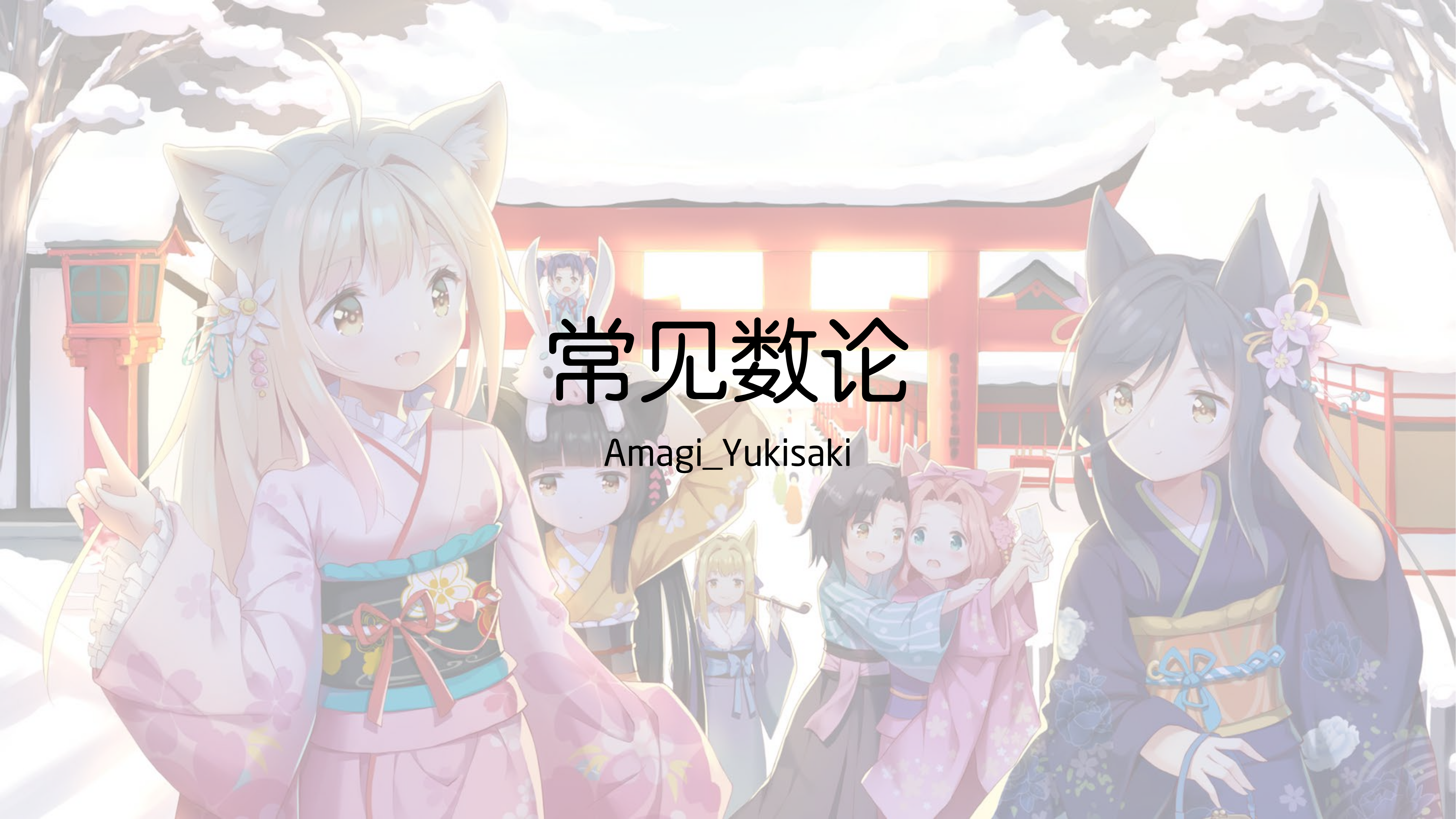
常见数论

Amagi_Yukisaki



关于这篇课件

- 这次我们要讲的内容有：欧几里得，素数相关，同余，逆元，容斥，线性筛法，卢卡斯定理，欧拉定理，欧拉函数，斐蜀定理，费马小定理……
- 不过大部分东西大家似乎都学过的说……
- 所以简单复习一下，然后直接做题
- 数论题目有一个特点就是出题方式灵活，形式多样，思维要求比较高，但是想到正解以后代码实现就相对简单了（我才没说多项式科技，斯特林反演之类奇奇怪怪的东西
- 总之希望大家多多思考，踊跃回答。不要出现几个大佬疯狂切题，其他人摸鱼划水的情况……
- 为了制裁抄代码的情况，本次课件中大部分代码示例将以$python3$语言给出。
- 如果听不懂也没有关系。至少不要留下一个：这道题上课讲了，很难，不可做的印象就好。
- 关于幻灯片背景？《此花亭奇谭》的同人插画，P站ID：66897076。（就知道你们想要这个
- 一个奇怪的链接：https://amagi.yukisaki.io/demo/Konohaha_Kitan

欧几里得，裴蜀定理

- 我们常见的欧几里得算法大概就是辗转相除及相关的思路。
- 其实就是普通的gcd算法和$exgcd$了。（拓展欧几里得？你们不需要知道
- 前者求出两个数的gcd，后者求出$a * x + b * y = gcd(a, b)$的一组解。（相当于构造出了裴蜀定理所要求的一组解）
- 它们的代码是这样的：

```python
def gcd(a, b):
    if b == 0:
        return a
    return gcd(b, a % b)

def exgcd(a, b): # return a tuple of (gcd, x, y), gcd = a * x + b * y
    if b == 0:
        return (a, 1, 0)
    temp = exgcd(b, a % b)
    return (temp[0], temp[2], temp[1] - a // b * temp[2])
```

素数相关，同余，逆元

- 众所周知，素数是质因子只有它自己的数（唯一分解后左右两边完全相同）
- 当a, b被m除有相同的余数的时候，称a, b对模m同余，记作$a \equiv b(mod\ m)$
- 当两个数a, b和一个模数m，满足$a * b \equiv 1(mod\ m)$时，则称a,b互为模m意义下的逆元。
- 逆元怎么求？对a, m做$exgcd$，求出的x与a的逆元在模m的意义下同余。
- 或者费马小定理+快速幂

线性求逆元

- 如何线性求出1到$m-1$中每个整数模质数m的逆元呢?
- 一个常见的方法:

```
inv[1] = 1
for i in range(2, m):
    inv[i] = (m - m // i) * inv[m % i] % m
```

- 一个不太正常的方法:
- 求出1到$m-1$的阶乘，快速幂求出$m-1$阶乘的逆元，倒退得到每个阶乘的逆元，之后$inv[i] = fac[i-1] * facInv[i] \% m$
- （如果能想到第一种的话不建议用第二种，无论代码量还是常数都存在劣势，唯一的好处就是连续寻址？）

同余方程

- 你有n个方程，每个形如：
- $x \equiv a_i \ (mod\ m_i)$
- 你要求出x在模$lcm(m_1, m_2, \dots, m_n)$下的值。
- 如果这些模数互质，中国剩余定理告诉你答案为：
- 定义$M = \prod_{i=1}^{n} m_i\,, M_i = \frac{M}{m_i}, t_i = M_i^{-1}(mod\ m_i)$
- $x \equiv \sum_{i=1}^{n} a_i t_i M_i \ (mod\ M)$
- 正确性并不显然，但我们可以用更一般的情况推出它

同余方程

- 考虑m_i不两两互质的情况，这时候我们可以每次将两个方程合并求出最终解。
- 考虑合并两个方程的情况：
- $x \equiv a_1 (mod\ m_1), x \equiv a_2 (mod\ m_2)$
- 那么，$x = a_1 + b_1 * m_1, x = a_2 + b_2 * m_2$
- 代换一下，得到：$b_1 * m_1 - b_2 * m_2 = a_2 - a_1$
- 求出$c_1 * m_1 - c_2 * m_2 = \gcd(m_1, m_2)$的一组解，构造以上方程的解，求出$x$即可。
- 如果$a_2 - a_1$模$\gcd(m_1, m_2)$的余数不为0，则无解喽。

卢卡斯定理

- 求$\binom{n}{m}\ mod\ p$，当p比较小的质数，而n,m非常大时，我们可以这样求解：
- $\binom{n}{m} \equiv \binom{n\%p}{m\%p}\binom{n/p}{m/p}(mod\ p)$
- 当p不是质数时，我们需要拓展卢卡斯定理。

拓展卢卡斯定理

- 当模数p不是质数时，我们先对p进行唯一分解。
- $p=\prod p_i^{k_i}$
- 求出答案模每个$p_i^{k_i}$的值，然后CRT合并。
- 求组合数本质就是求阶乘，故我们想办法求出$n!\ \%\ p_i^{k_i}$的值即可。
- 我们只需要知道： $n!$中p_i的次数（次幂数），记为b。
- 然后求出$\frac{n!}{p_i^b}\ \%\ p_i^{k_i}$的值，记为$a$。
- 注意到1到n这n个数在模$p_i^{k_i}$的意义下是循环的，所以对循环单独处理，最后余下的部分暴力即可。

拓展卢卡斯定理

- 分别求出三个阶乘的(a,b)后，怎么得到组合数?
- 对于p_i的次数，可以直接线性加减，如果答案中p_i的次数大于等于唯一分解中p_i的次数k_i，则答案模$p_i^{k_i}$的值为0。
- 否则对分数线下的两个阶乘的a值求逆元（用$exgcd$，因为a与$p_i^{k_i}$）互质所以一定有逆元，正常计算即可。
- （这种方法原生支持可重集的排列的计算）
- 代码？我其实是拒绝用$py3$写一遍这种东西的……

欧拉函数，欧拉定理，费马小定理

- 欧拉函数：$\varphi(n)$表示小于n的正整数中与n互质的数的个数。特别地，$\varphi(1)=1$。
- 欧拉定理：对于互质的正整数a,n，我们有：$a^{\varphi(n)}\equiv 1\ (mod\ n)$。
- 费马小定理：欧拉定理的特殊情况，对于质数p和不是p的倍数的正整数a，我们有：$a^{p-1}\equiv 1\ (mod\ p)$
- 所以当模数为质数时我们可以用费马小定理来求逆元：
- $inv(a)\ \equiv a^{p-2}\ (mod\ p)$

拉格朗日插值

- 对于不超过n次的多项式函数$f(x)$，我们若得到其上面$n+1$个点(x_i, y_i)，则对于定义域上任意x_0，可以快速求出$f(x_0)$的值。
- 具体方法为：
- $f(x) = \sum_{i=0}^{n} y_i \prod_{j=0, j\neq i}^{n} \frac{x-x_i}{x_j-x_i}$
- 如果预先给定的x_i为连续的一段区间的话，通过预处理$\prod_{j=0, j\neq i}^{n} x_j - x_i$的逆元，可以优化为O(n)
- （其实就是预处理阶乘的逆元和负阶乘的逆元啦）。

莫比乌斯反演

- 首先我们需要认识几个非常重要的数论函数：
- 欧拉函数(φ)：$\varphi(n)$表示$1\sim n$中与n互质的数的个数
- 莫比乌斯函数(μ)：$\mu(1)=1$，若$n(n>1)$含有多重质因数，则$\mu(n)=0$，否则 $\mu(n)=(-1)^r$，r表示n的质因数个数。
- 常函数(1)：$\forall n\in N$，$1(n)=1$
- 单位函数(id)：$\forall n\in N$，$id(n)=n$
- 狄利克雷卷积单位函数(ε)：$\varepsilon(1)=1$，$\forall n>1$，$\varepsilon(n)=0$
- 接着定义一个概念：迪利克雷卷积
- $(f*g)(n)=\sum_{d|n}f(n)g(\frac{n}{d})$

莫比乌斯反演

- 接着我们注意到:
- $\varphi\ x\ 1\ =\ id$（在n为质数的情况下显然，其余情况可用积性函数性质推出）
- $\mu\ x\ id\ =\ \varphi$（以后可以用上面证明）
- $\mu\ x\ 1\ =\ \varepsilon$
- $\varepsilon \times 1\ =\ 1$
- 注意前两条结论在以后的题目中会非常常用，即使暂时无法理解，强行背过也是个不错的选择

莫比乌斯反演

- 我们猜想对于一般的积性函数($f(x)$，$g(x)$)，是否存在：
- $f \times 1 = g \Leftrightarrow \mu \times g = f$
- 答案是肯定的，以下即为证明：

$\sum_{d|n} \mu(d) g(\frac{n}{d})$

$= \sum_{d|n} \mu(d) \sum_{d'|\frac{n}{d}} f(d')$ (带入已知)

$= \sum_{d'|n} f(d') \sum_{d|\frac{n}{d'}} \mu(d)$ (交换求和号)

$= \sum_{d'|n} f(d') \varepsilon(\frac{n}{d'})$ (代入 μ×1=ε)

$= f(n)$ (此时只有当 d'=n 时后面为 1)

$\sum_{d|n} f(d)$

$= \sum_{d|n} \sum_{d'|d} g(d') \mu(\frac{d}{d'})$ (带入已知)

$= \sum_{d'|n} g(d') \sum_{d'|d \& d|n} \mu(\frac{d}{d'})$ (提取公因子)

$= \sum_{d'|n} g(d') \sum_{d''|\frac{n}{d'}} \mu(d'')$ (变形)

$= \sum_{d'|n} g(d') \varepsilon(\frac{n}{d'})$ (代入 μ×1=ε)

$= g(n)$ (此时只有当 d'=n 时后面为 1)

部分细节

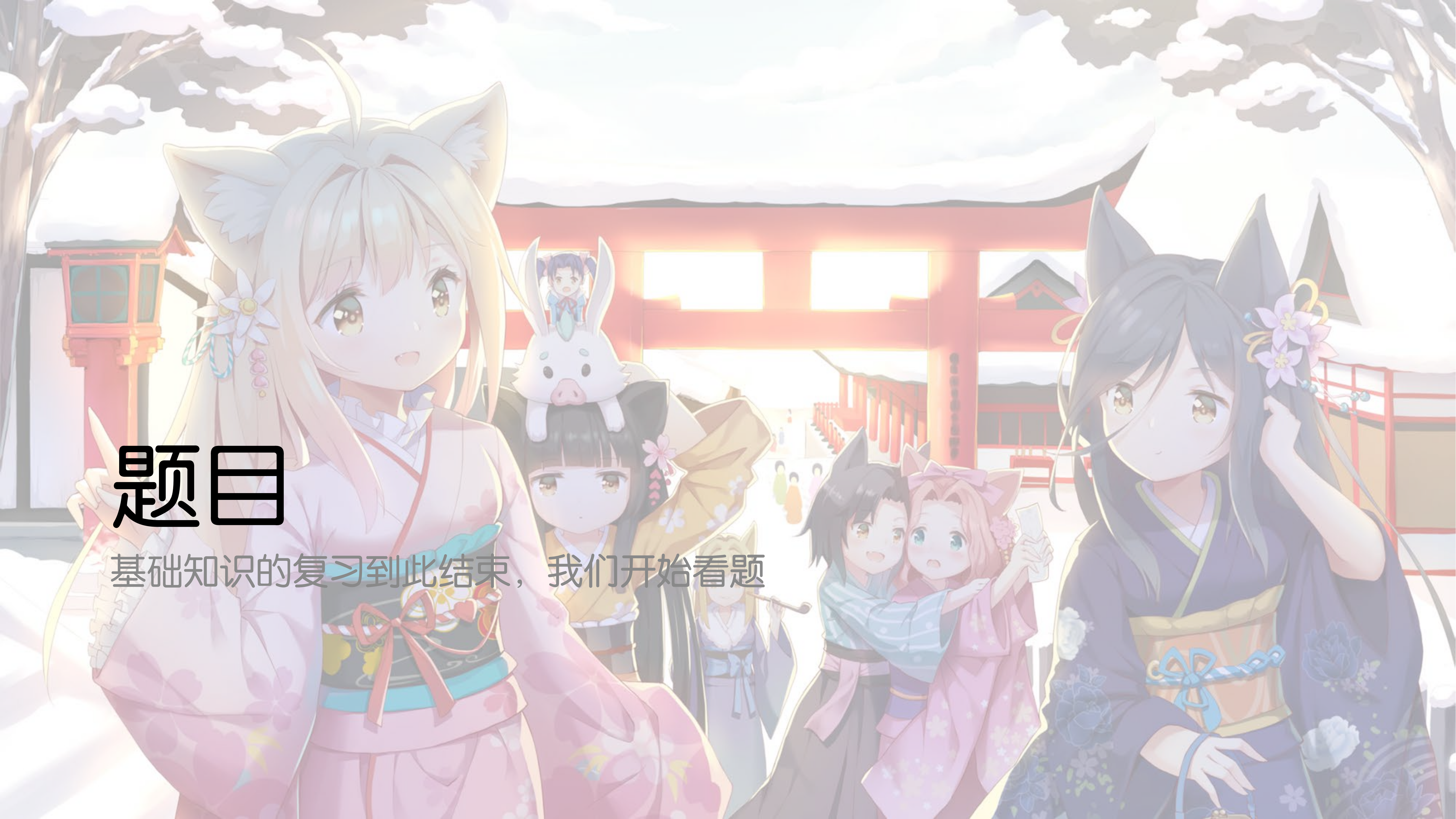
- 关于φ，μ如何线性求出?
- 请看以下代码：

```
for i in range(2, n + 1):
    if vis[i] == 0:
        cnt = cnt + 1
        prime[cnt] = i
        phi[i] = i - 1
        mu[i] = 1
    for j in range(1, cnt + 1):
        if i * prime[j] > n:
            break
        vis[i * prime[j]] = 1
        if i % prime[j]:
            phi[i * prime[j]] = phi[i] * (prime[j] - 1)
            mu[i * prime[j]] = -mu[i]
        else:
            phi[i * prime[j]] = phi[i] * prime[j]
            break
```

部分细节

- 数论分块：对于从1到n枚举的i，按照$\left\lfloor\frac{n}{i}\right\rfloor$的值进行分组的话，会分成$O(\sqrt{n})$组。（这种实例在题目中非常常见）
- 怎么实现?

```python
def calc(n):
    i = 1
    j = 1
    # ret = 0
    while i <= n:
        j = n // n // i
        i = j + 1
        # ret = ret + ......
    # return ret
```



题目

基础知识的复习到此结束，我们开始看题

BZOJ2142: 礼物

- 一年一度的圣诞节快要来到了。每年的圣诞节小E都会收到许多礼物，当然他也会送出许多礼物。不同的人物在小E心目中的重要性不同，在小E心中分量越重的人，收到的礼物会越多。小E从商店中购买了n件礼物，打算送给m个人，其中送给第i个人礼物数量为wi。请你帮忙计算出送礼物的方案数（两个方案被认为是不同的，当且仅当存在某个人在这两种方案中收到的礼物不同）。由于方案数可能会很大，你只需要输出模P后的结果。
- 对于100%的数据，$1 \leqslant n \leqslant 10^9$，$1 \leqslant m \leqslant 5$，$1 \leqslant pi^{ci} \leqslant 10^5$。

BZOJ2142: 礼物

- 一眼可重集排列。
- 模数不一定是质数？拓展卢卡斯定理板子题。
- 一下为某个丑陋的递归写法：
- inline lli exfac(lli n,lli pi,lli pk) {
-     if(n <= pi) return brute(n , pi , pk);
-     return exfac(n / pi , pi , pk) * brute(n , pi , pk) % pk;
- }
- brute(n , pi , pk)为暴力求余数部分的阶乘。

BZOJ5394: [Ynoi2016]炸脖龙

- 给一个长为n的序列，m次操作，每次操作：

1. 区间 $[l,r]$ 加 x
2. 对于区间 $[l,r]$ ，查询

$$a[l]^{a[l+1]^{a[l+2]\cdots\cdots}} \mod p$$

，一直到 a_r

请注意每次的模数不同。

- n , m <= 500000 , 序列中每个数在 [2,1e9] 内， p <= 2e7， 每次加上的数在 [0,2e9]

BZOJ5394: [Ynoi2016]炸脖龙

- 首先线性筛出$2*10^7$以内的φ函数。
- 发现对任何数n，不断取φ，在$O(logn)$的次数内会变为1。
- 所以我们维护区间加，单点查。
- 查询的时候暴力即可。

BZOJ4869: [Shoi2017]相逢是问候

- Informatikverbindetdichundmich.
- 信息将你我连结。B君希望以维护一个长度为n的数组，这个数组的下标为从1到n的正整数。一共有m个操作，可以分为两种：
- 0 l r表示将第l个到第r个数（al,al+1,...,ar）中的每一个数ai替换为c^ai，即c的ai次方，其中c是输入的一个常数，也就是执行赋值ai=c^ai
- 1 l r求第l个到第r个数的和，也就是输出：sigma(ai),l<=i<=rai因为
- 这个结果可能会很大，所以你只需要输出结果mod p的值即可。
- $1 \leqslant n \leqslant 50000, 1 \leqslant m \leqslant 50000, 1 \leqslant p \leqslant 100000000, 0 < c < p, 0 \leqslant ai < p$

BZOJ4869: [Shoi2017]相逢是问候

- 思路类似于上一题。
- 答案一定能看做是c的某次方。
- 逐级利用拓展欧拉定理，发现答案在多次操作以后会收敛。
- 暴力预处理出每一位上的数在多少次操作后会收敛。
- 线段树维护整个区间是否收敛，如果收敛则跳过该区间，没有收敛则递归下去。
- 查询就是线段树查询区间和啦。

BZOJ1257: [CQOI2007]余数之和

- 给出正整数n和k，计算j(n, k)=k mod 1 + k mod 2 + k mod 3 + … + k mod n的值
- 其中k mod i表示k除以i的余数。
- 例如j(5, 3)=3 mod 1 + 3 mod 2 + 3 mod 3 + 3 mod 4 + 3 mod 5=0+1+0+3+3=7

BZOJ1257: [CQOI2007]余数之和

- 我们先只考虑$n \le k$的情况。因为k取模一个大于k的数，余数永远为k。
- 注意到$k \bmod x = k - \left\lfloor \frac{k}{x} \right\rfloor * x$。
- 我们只需求出：$\sum_{i=1}^{\min(n,k)} \left\lfloor \frac{k}{i} \right\rfloor * i$。

```
def calc(k, n):
    i = 1
    j = 1
    ret = 0
    while i <= min(n, k):
        j = min(n, k // k // i)
        ret = ret + sum(i + 1, j) * (k // i)
        i = j + 1
    return ret
```

BZOJ2440: [中山市选2011]完全平方数

- 小 X 自幼就很喜欢数。但奇怪的是，他十分讨厌完全平方数。他觉得这些数看起来很令人难受。由此，他也讨厌所有是完全平方数的正整数倍的数。然而这丝毫不影响他对其他数的热爱。
  这天是小X的生日，小 W 想送一个数给他作为生日礼物。当然他不能送一个小X讨厌的数。他列出了所有小X不讨厌的数，然后选取了第 K个数送给了小X。小X很开心地收下了。
  然而现在小 W 却记不起送给小X的是哪个数了。你能帮他一下吗?
- 对于 100%的数据有 $1 \leqslant Ki \leqslant 10^9$， $T \leqslant 50$

BZOJ2440: [中山市选2011]完全平方数

- 我们可以猜测答案不会很大，不超过$2x$。
- 二分答案。考虑怎样计算1到n中有多少数有平方因数。
- $n-\sum_{i=2}^{\sqrt{n}}\left\lfloor\frac{n}{i*i}\right\rfloor$?
- 显然不对，这样会重。比如16,36就被减去了两次。
- 考虑容斥系数，有平方因子的i不应再被计算，其余的为$(-1)^{质因数次数+1}$，恰好为$-\mu$。
- $n-\sum_{i=2}^{\sqrt{n}}\left\lfloor\frac{n}{i*i}\right\rfloor*\mu(i)$，预处理$\mu$即可。

BZOJ1101: [POI2007]Zap

- FGD正在破解一段密码，他需要回答很多类似的问题：对于给定的整数a,b和d，有多少正整数对x,y，满足x<=a，y<=b，并且gcd(x,y)=d。作为FGD的同学，FGD希望得到你的帮助。
- 第一行包含一个正整数n，表示一共有n组询问。（1<=n<=50000）接下来n行，每行表示一个询问，每行三个正整数，分别为a,b,d。（1<=d<=a,b<=50000）

BZOJ1101: [POI2007]Zap

- 首先将x,y除以d，转化为gcd为1的问题。
- $\sum_{i=1}^{x}\sum_{j=1}^{y}\varepsilon(\gcd(i,j))$
- $=\sum_{i=1}^{x}\sum_{j=1}^{y}\sum_{d|i,d|j}\mu(d)$
- $=\sum_{d=1}^{\min(x,y)}\mu(d)*\left\lfloor\frac{x}{d}\right\rfloor\left\lfloor\frac{y}{d}\right\rfloor$
- 二维数论分块即可。就是求对于x,y的下一个位置的时候，两个取min 。

BZOJ2301: [HAOI2011]Problem b

- 对于给出的n个询问，每次求有多少个数对(x,y)，满足$a \leqslant x \leqslant b$，$c \leqslant y \leqslant d$，且$\gcd(x,y) = k$，$\gcd(x,y)$函数为x和y的最大公约数。
- 100%的数据满足：$1 \leqslant n \leqslant 50000$，$1 \leqslant a \leqslant b \leqslant 50000$，$1 \leqslant c \leqslant d \leqslant 50000$，$1 \leqslant k \leqslant 50000$

BZOJ2301: [HAOI2011]Problem b

- 和上一题基本相同。二维前缀和容斥一下就好了。
- $f([x_1, y_1], [x_2, y_2]) = f([1, y_1], [1, y_2]) - f([1, x_1 - 1], [1, y_2]) - f([1, y_1], [1, x_2 - 1]) + f([1, x_1 - 1], [1, x_2 - 1])$
- 大概能理解的吧……领会精神啦

BZOJ1319: Sgu261Discrete Roots

- 给出三个整数p,k,a,其中p为质数，求出所有满足x^k=a (mod p),0<=x<=p-1的x。
- 2<=p<=10^9, 2<=k<=100000, 0<=a.

BZOJ1319: Sgu261Discrete Roots

- 传说中的k次剩余?
- 不妨求出p的一个原根r，方程两边同时对r取离散模对数。
- 则方程可化为:
- $k * \log_r x \equiv \log_r a \ (mod\ p\ - 1)$
- 即:
- $k * \log_r x - t * (p - 1) = \log_r a$
- 利用$exgcd$求出上述方程中$\log_r x$的全部根即可。
- $\log_r a$怎么求？经典的$BSGS$问题。

BZOJ1319: Sgu261Discrete Roots

- 怎么求原根?
- 我们猜想p最小的原根不会很大，所以从1开始逐个枚举检查。
- 怎样检查?
- 原根r的性质为$r^{[1,p-1]}$遍历模p的整个剩余系（除了0）。
- 所以我们求出$p-1$所有的质因子，设该向量为a。
- 则$r^{(p-1)/a_i}$均不能为1，否则不可能遍历整个剩余系。
- （因为如果r_0不是原根，且$r_0^{k_0}, r_0^{p-1} \equiv 1(mod\ p)$，则可知$k_0|p-1$，而1的幂均为$1$，所以只用尝试$\frac{p-1}{a_i}$即可。）

BZOJ3944: Sum

给定一个正整数$N(N \le 2^{31}-1)$，

求$ans1=\sum_{i=1}^{n}\varphi(i), ans2=\sum_{i=1}^{n}\mu(i)$，多组询问

BZOJ3944: Sum

- 用这道题为大家科普一下杜教筛。
- 杜老师筛是一种用$O(n^{\frac{2}{3}})$的时间复杂度求积性函数f()前缀和的算法。
- 需要满足条件为：存在一个区间和可以$O(1)$求出的函数g，且函数$h = f * g$的前缀和可$O(1)$求出。

BZOJ3944: Sum

- 具体操作如下：
- $\sum_{i=1}^{n} h(i)$
- $= \sum_{i=1}^{n} \sum_{d|i} g(d) * f(\frac{i}{d})$
- $= \sum_{d=1}^{n} g(d) \sum_{i=1}^{\frac{n}{d}} f(i)$
- 所以，$g(1) \sum_{i=1}^{n} f(i) = \sum_{i=1}^{n} h(i) - \sum_{d=2}^{n} g(d) \sum_{i=1}^{\frac{n}{d}} f(i)$
- h的前缀和，g的区间和易得，后一项数论分块+递归子问题。
- 子问题中求前缀和的下标一定是n除某个数上取整的数。

BZOJ3944: Sum

- 分析一下复杂度，线性筛出$n^{\frac{2}{3}}$的f, g, h的前缀和。
- 之后只用递归$O(\sqrt[2]{n})$个子问题，每个子问题的复杂度为$O(\sqrt[2]{x})$。
- 积分一下，发现总复杂度为O($n^{\frac{2}{3}}$)。
- 其实你们不需要知道证明，只需要知道这东西的复杂度和怎么做题就行了。

BZOJ3944: Sum

- 对于$f = \varphi$，其对应的$g = 1$，$h = id$。
- 对于$f = \mu$，其对应的$g = 1$，$h = \varepsilon$。
- 对于$f = \varphi * id^k$，其对应的$g = id^k$，$h = id^{k+1}$
- id^k的前缀和和区间和怎么求?
- 区间可以转化为前缀之差。其前缀和显然是一个$k+1$次多项式。拉格朗日差值即可。（见$RYOI$？）
- 如果次数比较低的话，可以手玩。见BZOJ4916: 神犇和蒟蒻。

示例代码：

```python
def sumPhi(x):
    if x <= PRE:
        return prePhi[x]
    if vis[n // x]:
        return mem[n // x]
    vis[n // x] = 1
    ret = x * (x + 1) / 2
    i = 2
    while i <= x:
        j = x // x // i
        ret = ret - (j - i + 1) * sumPhi(x // i)
        i = j + 1
    mem[n // x] = ret
    return ret
```

国际惯例的

谢谢大家