517 编程具体数学中级班内部讲义

517 老师

2020 年 4 月 24 日

1 前言

1.1 为什么学

我们学习一样东西，首先要找到初心，就是为什么要学？不然学习过程中碰到了困难就会难以坚持，学习具体数学对青少年最大的帮助是能具象化的体会到数学与计算机科学的完美结合，学会充分使用计算资源与数学知识多角度分析并解决问题。并且，由于计算机丰富的表现力，学习者在计算机中应用数学方法的同时，也会更加爱上数学，更能体会那种对于一个问题的本质不断挖掘，不停探索的过程

1.2 如何学

任何一个事物，入门学习过程都应该提纲挈领，抓住主干，避免被细节所局限，尤其是现在信息大爆炸的时代，但是如今各种教材往往内容过于庞杂与形式化，忽略了互联网时代该有的用户体验，因此我们结合算法竞赛特有的生态环境，配合视频，课件，在线判题系统，课后答疑讨论机制等等，发明了一套基于编程基础的具体数学学习体系，可读性，可实操性都比传统的对着一本巨大的教材（五六百页甚至更多）吭哧吭哧埋头苦干要更加高效，更加有趣

1.3 注意事项

本课件是为 517 编程的学员专门制作，属于教学内容的一部分，讲义内容的拿捏，整理花费了很长时间，请尊重劳动成果，尊重价值守恒定律，避免外传，如您非 517 编程学员，看到本段文字，建议您立即停止观看，关注 517 编程微信公众号正常报课后再进行学习，以免给自己养成不劳而获的习气，对自身未来的发展造成不利的影响

目录

2 常见的一些符号表示的意义

1. $a|b$ 表示 a 是 b 的约数
2. $\lfloor \frac{a}{b} \rfloor$, 表示 a 除以 b 下取整
3. $\prod\limits_{i=1}^{n} p_i$ 表示 $p_1, p_2, ..., p_n$ 的乘积
4. $\sum\limits_{i=1}^{n} a_i$ 表示 $a_1, a_2, ..., a_n$ 的和
5. $a \equiv b \pmod p$ 表示 a 与 b 模 p 的结果是相同的
6. $\sum\limits_{d|n}$ 表示枚举 n 的所有因子 d
7. $f \times g(x)$ 表示 f 函数与 g 函数的狄利克雷卷积
8. $\int_a^b f(x)dx$ 表示定积分
9. $f(x) = [x = 1]$ 表示 $f(x)$ 只有在 $x = 1$ 的时候才为 1，其他情况为 0

3 数论函数

数论函数是定义域为正整数的函数, 又叫做算数函数

4 积性函数

积性函数, 我们在学欧拉函数的时候有所接触, 当时只是局限于欧拉函数, 现在我们开始更深入的学习积性函数相关知识

4.1 积性函数的定义

积性函数的定义

对于一个数论函数 $f(n)$, 如果 a, b 互质, $f(ab) = f(a)f(b)$, 那么 $f(n)$ 为积性函数. 特殊的, 如果对于任意的 a, b 都有 $f(ab) = f(a)f(b)$, 那么 $f(n)$ 为完全积性函数

4.2 常见的积性函数

注意：不同地方的符号表示可能不一样，此讲义以如下方式表示各个常见的积性函数

1. 欧拉函数 $\varphi(n)$
2. 莫比乌斯函数 $\mu(n)$
3. 幂函数 (完全积性) $f_k(n) = n^k$
4. 除数函数 $\sigma_x n = \sum\limits_{d|n} d^x$, 即所有因子的 x 次幂之和, 特殊的, 当 $x = 0$ 的时候表示约数的个数, 当 $x = 1$ 的时候表示约数之和
5. 单位函数 (完全积性)$id(n) = n$
6. 元函数 (完全积性) $e(n) = [n = 1]$
7. 恒等函数 (完全积性)$I(n) = 1$

4.3 利用线性筛预处理普通的积性函数

积性函数一般可以利用线性筛来预处理，比如莫比乌斯函数，欧拉函数等. 积性函数满足 $f(n)=\prod_{p_i} f(p_i^{k_i})$，线性筛的时候一般分为两种情况

1. 新增了一种质因子，这个比较简单，直接利用定义即可
2. 最小质因子的幂次加 1，根据具体题目具体分析

假设 f 函数为积性函数 $f(n)=\sum\limits_{d|n} d$(f 为因子之和函数), 线性筛的时候有两种情况

- 当 i 不是 p 的倍数, 那么 $f(i\times p)=f(i)\times f(p)$
- i 是 p 的倍数, 那么我们观察 $f(i\times p)$ 比 $f(i)$ 多了什么, 实际上是某个 $f(p_i^{k_i})$ 变成了 $f(p_i^{k_i+1})$, 在这里我们可以先除 $f(p_i^{k_i})$, 然后再乘 $f(p_i^{k_i+1})$.

预处理因子之和函数

```
int p[N], pn, f[N], g[N], t[N]; //g[i]是所有i的最小素因子的乘积p^k
bool flag[N]; //t[i]表示p^k的所有因子之和,p是最小的素因子
void init() {
  f[1] = 1; //初始化
  for (int i = 2; i < N; i++) {
    if (!flag[i]) {
      p[pn++] = i;
      f[i] = t[i] = i + 1;
      g[i] = i;
    }
    for (int j = 0; j < pn; j++) {
      flag[i * p[j]] = true;
      if (i % p[j] == 0) {//情况2
        g[i * p[j]] = g[i] * p[j];
        t[i * p[j]] = t[i] + g[i * p[j]];
        f[i * p[j]] = f[i] / t[i] * t[i*p[j]];
        break;
      } else {//情况1
        f[i * p[j]] = f[i] * f[p[j]];
        g[i * p[j]] = p[j];
        t[i * p[j]] = p[j] + 1;
      }
    }
  }
}
```

5 莫比乌斯函数

5.1 莫比乌斯函数的定义

莫比乌斯函数的定义

莫比乌斯函数 $\mu(m)$ 是根据 19 世纪的数学家奥古斯特-莫比乌斯命名的
定义 $\mu(1) = 1$
当 $x = p_1 p_2 ... p_n$, 且 p_i 为不同的质数 (也就是 x 由若干个幂次为 1 的质数相乘),
$\mu(x) = (-1)^n$ 其他情况 $\mu(x) = 0$

5.2 莫比乌斯函数的性质

1. $\sum\limits_{d|n} \mu(d) = [n == 1]$

   证明: 假设 $n = p_1^{a_1} \times p_2^{a_2} ... \times p_k^{a_k}$, n 的所有因子中, p_i 的幂次不为 1 的都不用考虑了, 因为 μ 值为 0. 那么接下来就是质数的幂次都为 1 的那些因子, 也就是每个 p_i 的幂次要么取 0 要么取 1, 一共有 2^k 种选法, 如果选了奇数个, μ 值就为-1, 否则为 1, 由组合数学的一个基本公式 $\sum\limits_{i=0}^{n} (-1)^i \binom{n}{i} = 0 (n > 1)$, 性质 1 得证

2. 莫比乌斯函数为积性函数

   证明: 证明莫比乌斯函数是积性函数只需要证明 $\mu(xy) = \mu(x) \times \mu(y)$ $(x, y$ 互质)

   - 首先如果 x, y 有一个数存在一个质因子的幂次大于 1, 那么左边和右边都等于 0, 等式成立
   - 当 x, y 都是由若干个质数的 1 次幂构成的时候, 假设 x 由 k_1 个质数构成, y 由 k_2 个质数构成, 那么左边等于 $(-1)^{(k_1+k_2)}$, 右边等于 $(-1)^{k_1} \times (-1)^{k_2}$

5.3 莫比乌斯函数的代码实现

利用莫比乌斯函数的积性性质可以使用线性筛来 $O(n)$ 预处理莫比乌斯函数

预处理莫比乌斯函数

```
const int N = 1000010;
bool flag[N];
int p[N],pn,mu[N];
void init() {
  pn=0;
  mu[1] = 1;
  for(int i = 2; i < N; i++) {
    if(!flag[i]) p[pn++] = i, mu[i] = -1;
    for(int j = 0; j < pn && i*p[j] < N; j++) {
      flag[i*p[j]] = true;
      if(i%p[j]==0) {
        mu[i*p[j]] = 0;
        break;
      }
      else
        mu[i*p[j]] = -mu[i];
    }
  }
}
```

6 莫比乌斯反演

6.1 一些观察

已知一个数论函数 $f(n) = \sum\limits_{d|n} g(d)$，假如已知 g, 很容易求出 f, 但是如果已知 f, 要求出 g, 这就需要莫比乌斯反演

首先我们观察几个小例子，比如

$f(1) = g(1)$

$f(2) = g(1) + g(2)$

$f(3) = g(1) + g(3)$
$f(4) = g(1) + g(2) + g(4)$
$f(6) = g(1) + g(2) + g(3) + g(6)$
$f(8) = g(1) + g(2) + g(4) + g(8)$
那么
$g(1) = f(1)$
$g(2) = f(2) - f(1)$
$g(3) = f(3) - f(1)$
$g(6) = f(6) - f(3) - f(2) + f(1)$
$g(8) = f(8) - f(4)$
$g(54) = f(54) - f(27) - f(18) + f(9)$
此时, 读者可以停下来尝试着自己找找规律, 下面我们直接给出结论

6.2 莫比乌斯反演公式

莫比乌斯反演公式

如果

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

那么

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

这两个式子是互为充分必要条件, 也就是说他们可以互相推导, 下面我们以上面推导到下面为例, 反向推导留给读者进行

已知 $f(n) = \sum_{d|n} g(d)$, 我们需要证明 $g(n) = \sum_{d|n} \mu(d) f(\frac{n}{d})$

首先利用 f 函数的定义可以将 f 函数用 g 函数来表示

$$\begin{aligned}
\sum_{d|n} \mu(d) f(\tfrac{n}{d}) &= \sum_{d|n} \mu(d) \sum_{k|\frac{n}{d}} g(k) \\
&= \sum_{d|n} \sum_{k|\frac{n}{d}} \mu(d) g(k) \\
&= \sum_{k|n} g(k) \sum_{d|\frac{n}{k}} \mu(d)
\end{aligned}$$

$$= g(n)$$

最后一步的推导是利用莫比乌斯函数的性质一, 我们发现只有在 $k = n$ 的时候 $\sum_{d|\frac{n}{k}} \mu(d)$ 才为 1, 其他情况为 0,

6.3 莫比乌斯反演的另外一种形式

莫比乌斯反演另一种形式

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

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

也就是枚举 n 的倍数去容斥

证明方法是同理的, 根据定义替代 $f(d)$, 然后进行化简

$$\sum_{n|d} f(d)\mu(\tfrac{d}{n}) = \sum_{n|d}\sum_{d|k} g(k)\mu(\tfrac{d}{n})$$

$$= \sum_{n|k} g(k) \sum_{p|\frac{k}{n}} \mu(p) \text{ (将 } g \text{ 提取出来, 先枚举 k, 然后令 } p = \tfrac{d}{n}\text{)}$$

$$= g(n)$$

上述式子当 $k = n$ 时, $\sum_{p|\frac{k}{n}} \mu(p) = 1$, 其他情况为 0, 所以

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

成立

7 狄利克雷卷积

7.1 定义

狄利克雷卷积定义

定义两个数论函数的狄利克雷卷积为

$$(f \times g)(n) = \sum_{d|n} f(d) \times g(\frac{n}{d})$$

也可以写成

$$(f \times g)(n) = \sum_{ab=n} f(a) \times g(b)$$

7.2 狄利克雷卷积的性质及其常见应用

1. 交换律: $f \times g = (g \times f)$
2. 结合律: $(f \times g) \times h = f \times (g \times h)$
3. 分配率: $(f + g) \times h = f \times h + g \times h$
4. 如果 $f(n), g(n)$ 都为积性函数, 那么他们的狄利克雷卷积也为积性函数
5. 任何数论函数与元函数的卷积等于函数自己, 即 $(f \times e)(n) = \sum_{d|n} f(d) \times e(\frac{n}{d}) = f(n)$
6. $(\mu \times I)(n) = \sum_{d|n} \mu(d) \times I(\frac{n}{d}) = \sum_{d|n} \mu(d) = e(n)$
7. $\varphi \times I(n) = \sum_{d|n} \varphi(d) \times I(\frac{n}{d}) = \sum_{d|n} \varphi(d) = n = id(n)$
8. $\varphi = id \times \mu$ (第七条的左右两边同乘 μ)

7.3 狄利克雷卷积的代码实现

复杂度:$O(nlogn)$

狄利克雷卷积

```
for (int i = 1; i <= n; i++) {
  for (int j = 1; i*j <= n; j++) {
    res[i * j] += f[i] * g[j];
  }
}
```

7.4 用狄利克雷卷积推导莫比乌斯反演

已知一个数论函数 $f(n) = \sum_{d|n} g(d)$，那么 $f = g \times I$ (g 函数与恒等函数 I 函数的卷积)，根据狄利克雷卷积的性质, 我们给你在两边同乘以 μ 函数, 那么可以得到

$$f \times \mu = g \times I \times \mu = g \times e = g$$

所以

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

启发:

狄利克雷卷积对原问题进行了更高层次的抽象, 换了个维度去思考原先的问题, 原问题变得非常的简单.

7.5 莫比乌斯函数与欧拉函数

我们回想一下欧拉函数的一个性质

$n = \sum_{d|n} \varphi(d)$

那么我们令 $f(n) = n$, 令 $g(n) = \varphi(n)$

于是我们可以反演出

$$\varphi(n) = \sum_{d|n} \mu(d) f(\frac{n}{d}) = \sum_{d|n} \mu(d) \frac{n}{d}$$

其实这也是狄利克雷卷积第 8 条应用

8 整除分块扩展知识

整除分块相信大家都不陌生，这里增加几个小结论，以后的学习可能会用到

关于整除分块还有三个结论

1. $\sum\limits_{i=1}^{n}\lfloor\frac{n}{i}\rfloor=2\sum\limits_{i=1}^{\lfloor\sqrt{n}\rfloor}\lfloor\frac{n}{i}\rfloor-(\lfloor\sqrt{n}\rfloor)^2$
2. $\sum\limits_{i=1}^{n}\lfloor\frac{n}{i}\rfloor=\sum\limits_{d=1}^{n}\sigma_0(d)$ ($\sigma_0(d)$ 表示 d 的因子个数)
3. $\sum\limits_{i=1}^{n}\lfloor\frac{n}{i}\rfloor=\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}[i\cdot j\le n]$

9 例题

9.1 例题一

9.1.1 问题描述

字符串的周期性

求有多少的长度为 n 的字符串 (由 26 个小写字母构成) 存在周期性, 如果一个字符串 s 存在周期性, 那么这个字符串可以由某个字符串 t 重复至少两次得到.
数据范围: $n \leq 100000$

9.1.2 问题求解

令 $f(n)$ 表示长度为 n 的字符串数量, 设 $g(d)$ 表示周期为 d 的非周期性字符串的数量，相当于 d 长度为一个周期，一共有 $\frac{n}{d}$ 个周期

因此 $f(n) = \sum\limits_{d|n} g(d)$，相当于所有字符串的总量等于各种不同周期的字符串数量之和

可以利用莫比乌斯反演求解 $g(n)$

$g(n) = \sum\limits_{d|n} f(d)\mu(\frac{n}{d})$

最后答案就是 $f(n) - g(n)$

9.2 例题二

9.2.1 问题描述

互质数对

求 $\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{m}[gcd(i,j)=1]$ $(n, m \leq 1e7)$, 即计算有多少对 (i,j) 互质, 假设 $n \leq m$

9.2.2 问题求解

这是一道非常经典的莫比乌斯反演练习题, 我们可以将 $[gcd(i,j) == 1]$ 利用莫比乌斯函数的性质 1 来替换

$\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{m}[gcd(i,j)=1] = \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{m}\sum\limits_{d|gcd(i,j)}^{n}\mu(d)$

$$= \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{d|i,d|j}^{n}\mu(d)$$

$$= \sum_{d=1}^{n}\mu(d)\sum_{d|i}^{n}\sum_{d|j}^{m}1$$

$$= \sum_{d=1}^{n}\mu(d)\lfloor\frac{n}{d}\rfloor\lfloor\frac{m}{d}\rfloor$$

预处理 μ 函数的前缀和之后利用整除分块可以在 $O(\sqrt{n}+\sqrt{m})$ 时间复杂度内搞定

例题 2 关键部分代码

```
//S[i]表示mu函数的前缀和
long long calc(int n,int m) {
  long long ans=0;
  if(n>m) swap(n,m);
  for(int i=1,j;i<=n;i=j+1) {
    j=min(n/(n/i),m/(m/i));
    ans += 1LL * (S[j]-S[i-1])*(n/i)*(m/i);
  }
  return ans;
}
```

9.3 例题三

9.3.1 问题描述

最小公倍数之和

求 $\sum_{i=1}^{n}\sum_{j=1}^{m}lcm(i,j)\ (n,m\le 1e7, n\le m)$

9.3.2 问题求解

$$\sum_{i=1}^{n}\sum_{j=1}^{m}lcm(i,j) = \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{ij}{gcd(i,j)}$$

$$
\begin{aligned}
&= \sum_{d=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{m} [gcd(i,j)=d] \frac{ij}{d} \\
&= \sum_{d=1}^{n} \sum_{i=1}^{\lfloor \frac{n}{d} \rfloor} \sum_{j=1}^{\lfloor \frac{m}{d} \rfloor} [gcd(i,j)=1] \frac{ijd^2}{d} \\
&= \sum_{d=1}^{n} d \sum_{i=1}^{\lfloor \frac{n}{d} \rfloor} \sum_{j=1}^{\lfloor \frac{m}{d} \rfloor} ij \sum_{k|i,k|j} \mu(k) \\
&= \sum_{d=1}^{n} d \sum_{k=1}^{\lfloor \frac{n}{d} \rfloor} \mu(k) \sum_{k|i}^{\lfloor \frac{n}{d} \rfloor} i \sum_{k|j}^{\lfloor \frac{m}{d} \rfloor} j \\
&= \sum_{d=1}^{n} d \sum_{k=1}^{\lfloor \frac{n}{d} \rfloor} k^2 \mu(k) \sum_{i=1}^{\lfloor \frac{n}{kd} \rfloor} i \sum_{j=1}^{\lfloor \frac{m}{kd} \rfloor} j
\end{aligned}
$$

分块套分块, 复杂度为 $O(n)$

10 常用技巧总结

1. 交换枚举的先后顺序，这个非常常用
2. 交换约数与倍数
3. 用莫比乌斯函数来替换 $[g == 1]$ 这样的式子
4. 替换和式中的标号
5. 整除分块加速计算
6. 枚举 gcd 的值 (如果式子里有 gcd(i,j) 的形式)，枚举 d 之后转换成 [gcd(i,j) == d] 的形式