基础数论 Day1

2023 SDUTACM 寒假集训

凌乱之风

山东理工大学

2023 年 2 月 2 日

目录

[前置] 数学符号介绍

数学符号

- $\sum$：求和符号，例如 $\sum_{i=1}^{n} i = \frac{n \times (n+1)}{2}$ 代表 $1+2+3+4+\cdots+n$
- $\prod$：连乘符号，例如 $\prod_{i=1}^{n} i = n!$ 代表 $1 \times 2 \times 3 \times 4 \times \cdots \times n$
- $\lfloor \frac{x}{y} \rfloor$：向下取整符号，例如 $\lfloor \frac{5}{2} \rfloor = 2$
- $\lceil \frac{x}{y} \rceil$：向上取整符号，例如 $\lceil \frac{5}{2} \rceil = 3$
- $[P] = \begin{cases} 1 & \text{if } P \text{ is true} \\ 0 & \text{Otherwise} \end{cases}$：艾弗森括号，例如 $[n=1]$ 只有 $n=1$ 时才取值为 1
- $x \mid y$：整除符号，表示 x 整除 y，也就是 x 是 y 的约数，例如 $2 \mid 4$

整除与约数

定义

- 设 $a, b \in \mathbb{Z}$ 且 $a \neq 0$，若 $\exists k \in \mathbb{Z}$ 满足 $b = k \times a$，那称为 b 被 a 整除。记作 $a \mid b$，否则 $a \nmid b$
- 若 $a \mid b$，则 b 是 a 的倍数，a 是 b 的约数。

最大公约数：Greatest Common Divisor，简称为 gcd

欧几里得算法

已知 $a, b \in \mathbb{Z}$，不妨设 $a > b$，那么 $a = k \times b + c$，则 $\gcd(a, b) = \gcd(b, c)$，$c = a \bmod b$

证明

- 设 $p = \gcd(a, b), q = \gcd(b, c)$
- $\because p \mid a, p \mid b$，$\because \dfrac{a}{p} - k \times \dfrac{b}{p} = \dfrac{c}{p}$，$\therefore p \mid c$，$\therefore p \mid q$
- $\because q \mid b, q \mid c$，$\because \dfrac{a}{q} = k \times \dfrac{b}{q} + \dfrac{c}{q}$，$\therefore q \mid a$，$\therefore q \mid p$
- $\because p \mid q, q \mid p$，$\therefore p = q$

参考代码 (时间复杂度 $O(\log n)$):

```
int gcd(int a, int b) {
    return b ? gcd(b, a % b) : a;
}
```

最小公倍数：Least Common Multiple，简称为 lcm

公式

$$\operatorname{lcm}(a, b) = \frac{a \times b}{\gcd(a, b)}$$

质数 (素数)，是指在大于 1 的自然数中，除了 1 和它本身以外不再有其他约数的自然数。

试除法判定质数

设判定的数为 n，若 x 是 n 的一个约数，那么 $\frac{n}{x}$ 也是 n 的一个约数。对于每一对约数，都在区间 $[1, \sqrt{n}]$ 中。

参考代码 (时间复杂度 $O(\sqrt{n})$):

```cpp
bool isPrime(int n) {
    if (n < 2) {
        return false;
    }
    for (int i = 2; i * i <= n; i ++) {
        if (n % i == 0) {
            return false;
        }
    }
    return true;
}
```

算术基本定理也称作唯一分解定理。

定义

任何一个正整数 n 都可以表示为 $n = p_1^{c_1} p_2^{c_2} \cdots p_k^{c_k}$，其中 p_i 为质数。

例

$6936 = 2^3 \times 3 \times 17$
$1200 = 2^4 \times 3 \times 5^2$
$5207 = 41 \times 127$
$114514 = 2 \times 31 \times 1847$

最小公倍数的证明

- 再次回顾最大公约数与最小公倍数，我们发现对于两个正整数 $a = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}, b = p_1^{\beta_1} p_2^{\beta_2} \cdots p_k^{\beta_k}$
- 最大公约数为 $\gcd(a, b) = p_1^{\min(\alpha_1,\beta_1)} p_2^{\min(\alpha_2,\beta_2)} \cdots p_k^{\min(\alpha_k,\beta_k)}$
- 最小公倍数为 $\mathsf{lcm}(a, b) = p_1^{\max(\alpha_1,\beta_1)} p_2^{\max(\alpha_2,\beta_2)} \cdots p_k^{\max(\alpha_k,\beta_k)}$
- 由于 $x + y = \min(x, y) + \max(x, y)$，所以 $\gcd(a, b) \times \mathsf{lcm}(a, b) = a \times b$

算术基本定理的两个推论

约数个数

$n = p_1^{c_1} p_2^{c_2} \cdots p_k^{c_k}$，那么根据乘法原理，从 p_1 中挑选有 $c_1 + 1$ 种选法，从 p_2 中挑选有 $c_2 + 1$ 种选法，$\cdots$
所以约数个数为

$$\prod_{i=1}^{k} (c_i + 1)$$

约数之和

$n = p_1^{c_1} p_2^{c_2} \cdots p_k^{c_k}$，将每个质因子看作一个多项式 $(1 + p_i + p_i^2 + \cdots + p_i^{c_i})$，将每个多项式乘起来，相当于挑选之后加起来。
所以约数之和为

$$\prod_{i=1}^{k} \sum_{j=0}^{c_i} p_i^j$$

分解质因数

枚举 $\sqrt{n}$ 内的质因子。

参考代码 (时间复杂度 $O(\sqrt{n})$):

```
vector<pair<int, int>> factor;
for (int i = 2; i * i <= n; i ++) {
    if (n % i == 0) {
        int cnt = 0;
        while (n % i == 0) {
            n /= i, cnt ++;
        }
        factor.push_back({i, cnt});
    }
}
if (n > 1) {
    factor.push_back({n, 1});
}
```

试除法求约数

因为约数总是成对出现，所以只需要枚举 $\sqrt{n}$ 内的约数。
参考代码 (时间复杂度 $O(\sqrt{n})$):

```
vector<int> factor;
for (int i = 1; i * i <= n; i ++) {
    if (n % i == 0) {
        factor.push_back(i);
        if (n / i != i) {
            factor.push_back(n / i);
        }
    }
}
```

欧拉函数

互质

$\forall a, b \in \mathbb{N}$，若 $\gcd(a, b) = 1$，则 a, b 互质。

定义

$1 \sim n$ 中与 n 互质的数的个数被称为欧拉函数，记作 $\varphi(n)$，即

$$\varphi(n) = \sum_{i=1}^{n} [\gcd(n, i) = 1]$$

计算式

若 $n = p_1^{c_1} p_2^{c_2} \cdots p_k^{c_k}$，则

$$\varphi(n) = n \times \prod_{i=1}^{k} \frac{p_i - 1}{p_i}$$

欧拉函数

试除法求欧拉函数值

类似于分解质因数，计算答案即可。
参考代码 (时间复杂度 $O(\sqrt{n})$):

```
int phi(int x) {
    int res = x;
    for (int i = 2; i * i <= x; i ++) {
        if (x % i == 0) {
            res = res / i * (i - 1);
            while (x % i == 0) {
                x /= i;
            }
        }
    }
    if (x > 1) {
        res = res / x * (x - 1);
    }
    return res;
}
```

积性函数

定义

$\forall n \in \mathbb{Z}^+$，有数论函数 $f(n)$ 满足 $\forall a, b, \gcd(a, b) = 1$ 使得 $f(a \times b) = f(a) \times f(b)$，则称 $f(n)$ 为积性函数。

常见的积性函数

- 欧拉函数：$\varphi(n) = \sum_{i=1}^{n} [\gcd(i, n) = 1]$
- 单位函数：$\varepsilon(n) = [n = 1]$
- 恒等函数：$Id(n) = n$
- 常数函数：$I(n) = 1$
- 约数个数函数：$d(n) = \sum_{i|n} 1$
- 约数和函数：$\sigma(n) = \sum_{d|n} d$

欧拉函数计算式的证明

引理 1

$\varphi(p^c) = p^c \times \dfrac{p-1}{p}$，其中 p 为质数。

证明：在 $1 \sim p^c$ 中，只有质因数中不包含 p 的数才与 p^c 互质，这样的数为 $p, 2 \times p, 3 \times p, \cdots, p^{c-1} \times p$，总共有 p^{c-1} 个数，所以

$$\varphi(p^c) = p^c - p^{c-1} = p^{c-1} \times (p-1) = p^c \times \frac{p-1}{p}$$

欧拉函数计算式的证明

证明

设 $n = \prod_{i=1}^{k} p_i^{c_i}$

$$\begin{aligned}
\varphi(n) &= \varphi\left(\prod_{i=1}^{k} p_i^{c_i}\right) &(1)\\
&= \prod_{i=1}^{k} \varphi(p_i^{c_i}) &(2)\\
&= \prod_{i=1}^{k} p_i^{c_i} \times \frac{p_i - 1}{p_i} &(3)\\
&= n \times \prod_{i=1}^{k} \frac{p_i - 1}{p_i} &(4)
\end{aligned}$$

朴素筛

算法

- 枚举 $i \in [2, n]$，标记 i 的倍数 $2 \times i, 3 \times i, \cdots$
- 没被标记过的数就是质数。

参考代码 (时间复杂度 $O(n \log n)$):

```
vector<int> primes;
vector<bool> st;
void sieve(int n) {
    st.resize(n + 1);
    for (int i = 2; i <= n; i ++) {
        if (!st[i]) {
            primes.push_back(i);
        }
        for (int j = 2 * i; j <= n; j += i) {
            st[j] = true;
        }
    }
}
```

朴素筛

时间复杂度证明

第 i 次循环执行 $\frac{n}{i}$ 次，总共执行 $\sum_{i=2}^{n}\frac{n}{i}=n\times\sum_{i=2}^{n}\frac{1}{i}$ 次，为调和级数 $\lim_{n\to\infty}\sum_{i=1}^{n}\frac{1}{i}=\ln n+C$，其中 $C\approx 0.57$，所以时间复杂度为 $O(n\ln n)\approx O(n\log n)$，这也是算法竞赛中常用的枚举倍数法。

埃氏筛

算法

- 对朴素筛进行优化，只需要考虑质数的倍数。
- 在枚举倍数时，因为 $i \times (2 \sim i-1)$ 已经被 $2 \sim i-1$ 标记过，所以可以从 i^2 开始。

参考代码 (时间复杂度 $O(n \log \log n)$):

```
vector<int> primes;
vector<bool> st;
void sieve(int n) {
    st.resize(n + 1);
    for (int i = 2; i <= n; i ++) {
        if (!st[i]) {
            primes.push_back(i);
            for (int j = i * i; j <= n; j += i) {
                st[j] = true;
            }
        }
    }
}
```

欧拉筛 (线性筛)

参考代码 (时间复杂度 $O(n)$):

```
vector<int> primes;
vector<bool> st;
void sieve(int n) {
    st.resize(n + 1);
    for (int i = 2; i <= n; i ++) {
        if (!st[i]) {
            primes.push_back(i);
        }
        for (auto p : primes) {
            if (i * p > n) {
                break;
            }
            st[i * p] = true;
            if (i % p == 0) {
                break;
            }
        }
    }
}
```

欧拉筛 (线性筛)

算法

- 考虑每个数的最小质因子。
- 维护一个质数数组，每次遍历 i 的时候从小到大枚举已有的每个质数 p
- 当 $p \mid i$ 时，p 一定是 i 的最小质因子，那么也一定是 $i \times p$ 的最小质因子。
- 当 $p \nmid i$ 时，那么 p 一定比 i 的最小质因子小，所以 p 也一定是 $i \times p$ 的最小质因子。

合理性

- 假设合数 n 的最小质因子为 p，那么 $\dfrac{n}{p}$ 一定会在 n 的前面枚举到。
- 此时再枚举质数时会枚举到 p，那么会把 $\dfrac{n}{p} \times p$ 标记。
- 所以每个数只会被它的最小质因子标记一次，总时间复杂度 $O(n)$

线性筛欧拉函数

算法

- 若 i 为质数，$\varphi(i) = i - 1$
- 当 $p \mid i$ 时，由于 p 已经是 i 的最小质因子，所以 $\varphi(i \times p) = \varphi(i) \times p$
- 当 $p \nmid i$ 时，此时 p 是 $i \times p$ 的一个新的最小质因子，所以 $\varphi(i \times p) = \varphi(i) \times (p - 1)$

参考代码 (时间复杂度 $O(n)$):

```
vector<int> primes, euler;
vector<bool> st;
void sieve(int n) {
    st.resize(n + 1), euler.resize(n + 1), euler[1] = 1;
    for (int i = 2; i <= n; i ++) {
        if (!st[i]) {
            euler[i] = i - 1, primes.push_back(i);
        }
        for (auto p : primes) {
            if (i * p > n) {
                break;
            }
            st[i * p] = true;
            if (i % p == 0) {
                euler[i * p] = euler[i] * p;
                break;
            }
            euler[i * p] = euler[i] * (p - 1);
        }
    }
}
```

思考题

- $\max\limits_{i=2}^{n} \frac{\varphi(i)}{i}, \min\limits_{i=2}^{n} \frac{\varphi(i)}{i}$
- $1 \le n \le 10^{18}$

习题 1. 质因数个数

题目链接：https://www.acwing.com/problem/content/description/4661/

习题 2. 阶乘分解

题目链接：https://www.acwing.com/problem/content/description/199/

习题 3. 模板题【线性筛求积性函数】

题目链接：https://ac.nowcoder.com/acm/contest/22769/A

习题 4. 华华给月月出题

题目链接：`https://ac.nowcoder.com/acm/contest/22769/B`

习题 5.Number Factorization

题目链接：`https://codeforces.com/contest/1787/problem/B`

习题 6. 来点 gcd

题目链接：`https://ac.nowcoder.com/acm/problem/229589`

习题 7.Ginger 的购物计划

题目链接：`https://codeforces.com/gym/103800/problem/A`
SDUTOJ 链接：`http://acm.sdut.edu.cn/onlinejudge3/problems/4899`

习题 8.[SDOI2012] Longge 的问题

题目链接：`https://www.luogu.com.cn/problem/P2303`

Thanks!