AcWing 算法基础、提高课数学知识

廖涛

2022 年 4 月 15 日

目录

1 质数 5

1.1 判定质数 5

1.2 分解质因数 6

1.3 质数筛 7

1.3.1 埃氏筛法 7

1.3.2 欧拉筛法 (线形筛法) 8

2 约数 9

2.1 试除法求约数 9

2.2 约数个数 9

2.3 约数之和 9

2.4 最小公约数 9

3 欧拉函数 11

4 快速幂 13

4.1 快速幂原理 13

4.2 快速幂求逆元 13

5 扩展欧几里得算法（粉碎机） 15

5.1 扩展欧几里得算法原理 15

5.2 扩展欧几里得算法求逆元 15

6 中国剩余定理（孙子定理） 17

7 组合数 19

8 容斥原理 **21**

9 博弈论 **23**

Chapter 1

质数

$\forall n \in [(1,+\infty) \cap Z^{+}]$，若 n 只有 1 和 n 两个约数，那么 n 为质数（素数），否则为合数。

0、1 以及负整数既不是质数也不是合数。

1.1 判定质数

```
bool is_prime(int n)
{
    if(n < 2) return false;

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

1. 是否是大于等于 2 的整数，2. 从 2 n-1 中是否有 n 的约数。

$d \mid n$，则必有 $\frac{n}{d} \mid n$，故不需要枚举到 $n-1$，即枚举每组约数的最小值即可。

时间复杂度为 $O(\sqrt{n})$

1.2 分解质因数

```
void divide(int n)
{
    for(int i = 2; i <= n / i; i ++)
    {
        int cnt = 0;
        if(n % i == 0)
        {
            while(n % i == 0)
            {
                n /= i;
                cnt ++;
            }
            printf("%d %d\n", i, cnt);
        }

    }

    if(n > 1) printf("%d 1\n", n);
    puts("");
}
```

从 2 开始枚举到 $\sqrt{n}$，每找到一个质因子则将 n 中的该质因子除尽，最后判断 n 是否大于 1，若大于 1，则此时的 n 也是所求质因子之一。

Tips1：所有合数因子在被枚举到之前会被更小的质数因子除尽。故枚举到的合数必定不会满足 $n \bmod i = 0$。eg：$n = 24$，8 这个合数因子必然会被之前出现的 2 除尽（除 3 次）。

Tips2：n 最多只存在 1 个大于等于 $\sqrt{n}$ 的质因子，故只需要枚举到 $\sqrt{n}$ 再判断最后的 n 情况如何即可。简单证明，假设存在

$$a \mid n, b \mid n, a > \sqrt{n}, b > \sqrt{n}$$

前两个条件可得 $a \times b \mid n$，最后一个条件可得 $a \times b > n$，相悖，故假设不成立，故 n 最多只存在 1 个大于等于 $\sqrt{n}$ 的质因子。

Tips3：当 $n = 2^k$ 时，第一次枚举即能把 n 除尽，此时的时间复杂度为 $\log_2 n$，最坏情况则是需要枚举到 $\sqrt{n}$，故该方法的时间复杂度为为 $O(\log_2 n)$ 与 $O(\sqrt{n})$ 之间。

1.3 质数筛

1.3.1 埃氏筛法

```
void get_primes(int n)
{
    for(int i = 2; i <= n; i ++)
    {
        if(st[i]) continue;

        primes[cnt ++] = i;
        for(int j = 2 * i; j <= n; j += i)
        {
            st[j] = true;
        }
    }
}
```

从 2 枚举到 n，即能找到 n 以内的所有质数。每次枚举到的 i 若 st[i] 为 false 则说明 i 为质数，记录到 primes[] 中，并筛掉其倍数。否则 i 为合数，直接跳过本次循环。

这里说明为什么不需要用合数筛掉其他数，当 st[i] 为 true 时证明之前已经找到了 i 的一个或多个质因子，i 的倍数也必定是这些质因子的倍数，所以当 i 为合数时，i 的倍数已经在之前被其质因子筛掉，不需要再筛。

在枚举到 i 时，2 $i-1$ 的所有数都已经枚举过，若 st[i] 为 false 则说明 2 $i-1$ 均不是 i 的因数，则 i 为质数（素数），否则为合数。

时间复杂度分析，当 $i=2$ 时内层循环次数为 $\frac{n}{2}$，当 $i=3$ 时为 $\frac{n}{3}$...，这里我们假设合数也进行内层循环，则内层循环的次数和为

$$\frac{n}{2}+\frac{n}{3}+\frac{n}{4}+...+\frac{n}{n}=n\times(\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+...+\frac{1}{n})$$

调和级数

$$\lim_{n\to\infty}\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+...+\frac{1}{n}=\ln n$$

故内层循环的次数小于等于 $n\ln n$ 小于 $nlog_2n$，故可以认为埃氏筛法的时间复杂度为

$$O(n\log_2 n)$$

1.3.2 欧拉筛法 (线形筛法)

```
void get_primes(int n)
{
    for(int i = 2; i <= n; i ++)
    {
        if(!st[i]) primes[cnt ++] = i;

        for(int j = 0; primes[j] <= n / i; j ++)
        {
            st[primes[j] * i] = true;
            if(i % primes[j] == 0) break;
        }
    }
}
```

欧拉筛法的一个特点之一就是每个合数都是用其最小的质因子筛掉。

这里重点讲内层循环。首先无论 i 是否为质数,都会进内层循环。$primes[]$ 中小的数一定在前面。

当 $i \bmod primes[j]! = 0$ 时, $primes[j]$ 的不是 i 的因子, 则 $primes[j] * i$ 的最小质因子就是 $primes[j]$。

当 $i \bmod primes[j] == 0$ 时，$primes[j]$ 是 i 的最小质因子，也是 $primes[j] * i$ 的最小质因子。但当 j ++ 后，$primes[j+1] * i$ 有更小的质因子 $primes[j]$（$primes[j] \mid i$）这里就不满足用最小质因子筛去合数的要求，所以要 break。

Chapter 2

约数

2.1 试除法求约数

2.2 约数个数

2.3 约数之和

2.4 最小公约数

Chapter 3

欧拉函数

Chapter 4

快速幂

4.1 快速幂原理

4.2 快速幂求逆元

Chapter 5

扩展欧几里得算法（粉碎机）

5.1 扩展欧几里得算法原理

5.2 扩展欧几里得算法求逆元

Chapter 6

中国剩余定理（孙子定理）

Chapter 7

组合数

Chapter 8

容斥原理

Chapter 9

博弈论