质数

李淳风

长郡中学

2024 年 8 月 1 日

定义

若一个大于 1 的正整数无法被除了 1 和它自身之外的任何自然数整除，则称该数为质数（或素数），否则称该正整数为合数。
虽然我们可以把这个定义应用到负数上，但一般我们只考虑正数的情况。
在整个自然数集合中，质数的数量不多，分布比较稀疏。对于一个足够大的整数 n，不超过 n 的质数大约有 $n/\ln n$ 个。

质数的判定

若一个正整数 n 是合数，则存在一个能整除 n 的数 T，其中 $2 \le T \le \sqrt{n}$。

证明：由于 n 是合数，那么 $\exists T$ 使得 $T|n$。设 $M = n/T$，则有 $M|n$ 且 $MT = n$。使用反证法，若 $T, M > \sqrt{n}$，则 $TM > n$，矛盾。故 M, T 中至少有一个不超过 $\sqrt{n}$。

根据上述结论，我们只需要扫描 $2 \sim \sqrt{n}$ 之间的所有整数，依次它们判断能否整除 n。若都不能整除，则 n 是质数，否则 n 是合数。当然，需要特判 $n = 0, 1$。该算法的时间复杂度为 $O(\sqrt{n})$。

质数的筛选

给定一个整数 n，求出 $1 \sim n$ 之间的所有质数，称为质数的筛选问题。Eratosthenes 筛法基于这样的想法：任意整数 x 的倍数，$2x, 3x, \cdots$ 都不是质数。我们可以从 2 开始，由小到大扫描每个数 x，把它的倍数 $2x, 3x, \cdots, \lfloor n/x \rfloor * x$ 标记为合数。当扫描到一个数时，若它尚未被标记，则它不能被 $2 \sim x-1$ 之间的任何数整除，该数就是质数。
另外，我们可以发现，2 和 3 都会把 6 标记为合数。实际上，小于 x^2 的 x 的倍数在扫描更小的数时就已经被标记过了。例如，当 $x > 3$ 时，$3x$ 在扫描 3 的倍数的时候就已经被标记过了。因此，我们只需要对所有是质数的 x 标记倍数，并且从 x^2 开始标记即可。
埃氏筛的时间复杂度为 $O(\sum_{p \le n, p \in Prime} \frac{n}{p}) = O(n \log\log n)$。

线性筛

埃氏筛的复杂度不是线性，是因为哪怕在优化之后，一个合数仍然会被多次标记，例如 12 会被 2 和 3 标记。其根本原因是我们没有唯一确定 12 的表示方式，其既可以被表示为 2×6，又可以被表示为 3×4。线性筛法通过“从大到小累积质因子”的方式来标记每个合数，即让 12 只有 $3 \times 2 \times 2$ 一种产生方式。

我们用数组 v 来记录每个数的最小质因子，我们按照如下步骤来计算：

- 依次考虑 $2 \sim n$ 之间的每个数 i。
- 若 $v_i = 0$，说明 i 是质数，把它用一个数组记录下来。
- 扫描不大于 v_i 的每个质数 p，令 $v_{i*p} = p$。也就是在 i 的基础上累积一个质因子 p。由于 $p \leq v_i$，所以 p 就是 $i * p$ 的最小质因子。

线性筛

```
void primes(int n){
  memset(v,0,sizeof(v));  //最小质因子
  m=0;    //质数数量
  for(int i=2;i<=n;i++){
    if(v[i]==0) {v[i]=i;prime[++m]=i;}    //i 是质数
    //给当前的数 i 乘上一个质因子
    for(int j=1;j<=m;j++){
      //i 有比 prime[j] 更小的质因子，或超出 n 的范围，停止循环
      if(prime[j]>v[i] || prime[j]>n/i) break;
      //prime[j] 是合数 i*prime[j] 的最小质因子
      v[i*prime[j]]=prime[j];
    }
  }
}
```

质因数分解

算数基本定理：任何一个大于 1 的正整数都能唯一分解为有限个质数的乘积，可以写作：

$$n = p_1^{c_1} p_2^{c_2} \cdots p_m^{c_m}$$

其中 c_i 都是正整数，p_i 都是质数，且满足 $p_1 < p_2 < \cdots < p_m$。
结合质数判定的“试除法”和质数筛选的埃氏筛，我们可以扫描 $2 \sim \sqrt{n}$ 的每个数 d，若 d 能整除 n，则从 n 中除掉所有的因子 d，同时累计除去的 d 的个数。
因为一个合数的因子一定在扫描到它这个数之前就一定从 n 中被除掉了，所以在上述过程中能整除 n 的一定是质数。最终我们就得到了质因数分解的结果，时间复杂度为 $O(\sqrt{n})$。如果已经有了一个质数表的话，那么可以只枚举质数进行试除，复杂度变为 $O(\frac{\sqrt{n}}{\ln \sqrt{n}})$。
特别地，如果 n 没有被任何 $2 \sim \sqrt{n}$ 的数整除，则 n 是质数，无需分解。

例题

Prime Distance

给定两个整数 $L, R(1 \leq L \leq R \leq 2^31, R - l \leq 10^6)$，求闭区间 $[L, R]$ 中相邻两个质数的差最大时多少，输出这两个质数。

例题

Prime Distance

给定两个整数 $L, R(1 \leq L \leq R \leq 2^31, R - l \leq 10^6)$，求闭区间 $[L, R]$ 中相邻两个质数的差最大时多少，输出这两个质数。

$[L, R]$ 的范围很大，我们无法直接求出 $[1, R]$ 区间内的所有质数；但是 $R - L$ 的值很小，并且任何一个合数 n 必定包含一个不超过 $\sqrt{n}$ 的质因子。
所以我们只需要用筛法求出 $2 \sim \sqrt{n}$ 之间的所有质数，并对于每个质数 p，把 $[L, R]$ 中能被 p 整除的数打上标记，表示这个数是合数。最终没有被标记的数就是质数。对相邻的质数两两比较，找出差最大的即可。
时间复杂度为：

$$O(\sum_{p \leq \sqrt{R}, p \in Prime} \frac{R - L}{p}) = O(\sqrt{R} \log\log\sqrt{R} + (R - L)\log\log R)$$

例题

阶乘分解

给定整数 $n(1 \leq n \leq 10^6)$，试把阶乘 $n!$ 分解质因数，按照算数基本定理的形式输出分解结果中的 p_i 和 c_i 即可。

例题

阶乘分解

给定整数 $n(1 \le n \le 10^6)$，试把阶乘 $n!$ 分解质因数，按照算数基本定理的形式输出分解结果中的 p_i 和 c_i 即可。

若把 $1 \sim n$ 每个数都分别分解质因数，在把结果合并，时间复杂度过高，是 $O(n\sqrt{n})$。但我们注意到，$n!$ 的每个质因子都不会超过 n，我们可以先筛出 $1 \sim n$ 的每个质数 p，然后再考虑 $n!$ 中一共包含多少个质因子 p。

$n!$ 中个质因子 p 的个数，等于 $1 \sim n$ 每个数包含质因子 p 的个数之和。在 $1 \sim n$ 中，至少包含一个质因子 p 的数有 $\lfloor n/p \rfloor$ 个。而至少包含两个质因子 p 的数有 $\lfloor n/p^2 \rfloor$ 个。不过其中的一个质因子已经在之前计算过了，所以不需要再乘以二。

综上所述，$n!$ 中质因子 p 的个数为：$\sum_{p^k \le n} \lfloor \frac{n}{p^k} \rfloor$

$1 \sim n$ 中质数个数约为 $O(n/\log n)$ 个，对于每个 p，我们需要 $O(\log n)$ 的时间进行计算，故总复杂度为 $O(n)$。