# 哥德巴赫猜想的验证

1900012476 夏罗生

## 1 思路

哥德巴赫猜想：一个大于 2 的偶数 $n$ 可以分解为两个素数之和。显然，其中一个素数一定 $\le \frac{n}{2}$，另一个素数一定 $\ge \frac{n}{2}$。

设偶数 $n$ 的范围为 $[2, MAX]$，最快速的判断方式是建立 $[2, MAX]$ 范围内的素数表，然后对于给定的 $n$，遍历素数表中 $[0, \frac{n}{2}]$ 中的每一项 $i$，查找 $\frac{n}{2} - i$ 是否也在表内，由于素数表已排好序，所以使用二分查找时间复杂度为 $O(log_2(n))$。

但是这个方法对于空间的要求很大，题目要求在无符号 32bit 整数范围内验证，即 $MAX = 2^{32}$，素数表长度约为 $\frac{MAX}{\ln MAX} = 1.9 * 10^8 \approx 2 * 10^8$，每一个素数使用 32bit 大小的 `int` 数据类型存储，即一个素数大小为 4B。

则素数表内存大小为 $\frac{2*10^8*4B}{10^6} = 800MB$，有点太大了，想办法减少一点空间。注意到，在进行验证时，其中一个素数一定 $\le \frac{n}{2}$。则可以只构建 $[0, \frac{n}{2}]$ 范围内的素数表，遍历这个素数表中的每一项 $i$，使用素性测试快速判断 $\frac{n}{2} - i$ 是否为素数。此时所需要的空间大概为 $400MB$，勉强能够接受。

相当于使用素性检验替代二分查找，用时间和准确性换取空间。

## 2 素数表

基础的素数筛算法有

- 埃氏筛，时间复杂度 $O(n \log\log n)$
- 欧拉筛，时间复杂度 $O(n)$

虽然暴力的埃氏筛和线性筛比起来是慢的，但是埃氏筛的潜力是无穷的，可以进行高度优化，如 GitHub 上的 kimwalisch/primesieve: 🚀 Fast prime number generator (github.com) 项目，

在 16 个线程下筛 $10^9$ 以内的质数只需要 $22ms$ 左右。下面介绍一下这个项目的一些优化方法。

### 2.1 分块筛选 Segmentation

显然，要找到直到 $n$ 为止的所有素数，仅对不超过 $\sqrt{n}$ 的素数进行筛选就足够了。即不需要保留整个 `is_prime[1...n]` 数组，而是 `prime[1...sqrt(n)]`。并将整个范围分成大小为 $S$ 的若干块，依次对每个块进行筛选，则在筛法运行中所需的内存降低为 $O(\sqrt{n} + S)$。

primesieve 算法将块的大小 $S$，设置为 CPU 的 L1 或 L2 缓存的大小，最大程度利用 CPU 的能力。

### 2.2 轮式优化 Wheel Factorization

Wheel Factorization 是使用前几个小素数加快素数判断过程，相当于剪枝的过程。下面举几个简单的优化例子帮助理解：

- 使用小素数 2，如果一个数能被 2 整除，那么一定不是素数。
- 使用小素数 $\{2, 3\}$，最大公约数为 6。一个数 $\mod 6$ 后，如果余数能被上述 2 个素数中的任何一个整除，那么一定不是素数。
- 使用小素数 $\{2, 3, 5, 7\}$，最大公约数为 210。一个数 $\mod 210$ 后，如果余数能被上述 4 个素数中的任何一个整除，那么一定不是素数。根据容斥原理不难算出，对 210 取模之后的 210种余数，只有 48 种余数还有可能是质数。这其实就带来特别大的优化了。

primesieve 算法就是在不同区间规模，分别针对 $\mod 30$ 和 $\mod 210$ 进行筛选。

### 2.3 其他优化

包括但不限于：

- 提前筛选小于 100 的素数。
- 使用多线程计算。
- 使用自定义大小的内存池

## 3 素性测试

### 3.1 定义

素性测试 (Primality test）是一类在不对给定数字进行素数分解（prime factorization）的情况下，测试其是否为素数的算法。

素性测试有两种：

1. 确定性测试：绝对确定一个数是否为素数。常见示例包括 Lucas-Lehmer 测试和椭圆曲线素性证明。
2. 概率性测试：通常比确定性测试快很多，但有可能（尽管概率很小）错误地将合数识别为素数（尽管反之则不会）。因此，通过概率素性测试的数字被称为可能素数，直到它们的素数可以被确定性地证明。而通过测试但实际上是合数的数字则被称为伪素数。有许多特定类型的伪素数，最常见的是费马伪素数，它们是满足费马小定理的合数。

### 3.2 Fermat 素性检验

费马小定理：若 $p$ 为素数，若 $a$ 与 $p$ 互质，则 $a^{p-1} \equiv 1 \mod p$

反过来，如果存在某个 $a^{p-1} \equiv 1 \mod p$，是否能够判定 $p$ 是素数呢？并不行，但是一个合数是费马伪素数的概率并不高，所以我们可以多测试几个 $a$，如果多组测试都成立，那么 $p$ 很可能是素数了。

### 3.3 Miller-Rabin 素性测试

Miller-Rabin 素性测试（Miller–Rabin primality test）是进阶的素数判定方法。

二次探测定理：对于素数 p，若 $x^2 \equiv 1 \mod p$，则小于 p 的解只有两个，$x_1 = 1, x_2 = -1$

对于待检验的奇数 $x$，可以把 $a^{x-1}$ 表示为 $a^{2^r d}$。对于 $a^d, a^{2d}, a^{4d}, \ldots, a^{2^r d}$ 这一串数字，后一个都是前一个数的平方。

我们已经知道对于奇素数 $x$，$a^{2^r d} \equiv 1 \mod x$，则 $a^{2^{r-1} d}$ 在 $\mod p$ 的情况下一定是 1 或者 -1。这意味着，如果 x 是奇素数，$a^d, a^{2d}, a^{4d}, \ldots, a^{2^r d}$ 这一串数字 $\mod p$，要么全是 1，要么中间某个数是 $-1$，并且从那往后全是 1。

反过来说，如果满足这个条件，$x$ 就很可能是素数。于是我们选择一些 $a$ 并验证是否符合条件即可，这就是 Miller-Rabin 素性测试。

米勒-拉宾素性检验是一种概率算法，但通过选取适合的底数，可以保证绝对正确。

在 `int` 范围内，选取 $2, 7, 61$ 三个数作为底数可以保证绝对正确。

在 `long long` 范围内，选取 $2, 325, 9357, 28178, 450775, 9780504, 1795265022$ 七个数作为底数可以保证绝对正确。