Integer Factorization Algorithms

Yiheng Su

May 2022

1 Introduction

Integer factorization is the decomposition of a composite number into a product of smaller integers. “Can integer factorization be solved in polynomial times?” It is a famous unsolved problem in computer science and mathematics. Till today, no classical algorithm[1] is known that can factor integers in polynomial time. However, neither the non-existence of such algorithms has been proved.

In this article, I will introduce five integer factorization algorithms and compare the runtimes between them. Firstly, I will describe the most straightforward algorithm, the trial division, and the wheel factorization improvement. Secondly, I will describe Fermat's factorization method and its improvement, Kraitchik-Fermat's factorization method. Thirdly, I will focus on a more intriguing and complicated algorithm called Pollard's $p-1$ factorization. Lastly, I will compare the runtimes between different algorithms on factoring various composite integers.

2 Definitions

1. **Semiprime**: a semiprime is a natural number that is the product of exactly two prime numbers.

2. **d-digit number**: An d digit number is a positive number with exactly d digits. For example, 1234 is a 4-digit number.

3. **Trivial and Nontrivial Factor**: For any number n, trivial factors are: ± 1 and $\pm n$. Any other factor, if it exists, is nontrivial factor.

[1] We do not consider any quantum integer factorization algorithm in this article.

3 Algorithms

3.1 Trial Division

Trial division is the most straightforward algorithm for factoring an integer. In the trial division, we need to divide a number n successively by the element in the sequence $S = \{2, 3, 4, ..., \lfloor\sqrt{n}\rfloor\}$ until we find a divisor. The following theorem guarantees that the upper bound for the testing numbers is $\lfloor\sqrt{n}\rfloor$.

Theorem 3.1. *If $n = pq$, p and q are nontrivial factors of n and $p \leq q$, then $p \leq \sqrt{n}$ [**?**].*

Proof. Suppose, by contradiction that, $p > \sqrt{n}$. Then, $q \geq p > \sqrt{n}$. Hence, $pq > \sqrt{n} \cdot \sqrt{n} = n$. This contradicts with $n = pq$. □

According to Theorem 3.1, we can design a trial division algorithm by testing all numbers from 2 to $\lfloor\sqrt{n}\rfloor$.

3.2 Trial Division: Algorithm

Algorithm 1 Trial Division

Require: Give a composite number $n \in \mathbb{N}$.
Ensure: Find a nontrivial factor of n.

```
f ← 2
while f * f < n do
    if n ≡ 0 (mod f) then
        return f
    end if
    f ← f + 1
end while
return n
```

3.3 Wheel Factorization

Wheel factorization is an improvement of the trial division. In the trial division, we need to divide a number n successively by the element in the sequence $S = \{2, 3, 4, ..., \lfloor n\rfloor\}$ until we find a divisor. For the wheel factorization, we create a sequence of the first few primes, called the **basis**, and then we generate a sequence, called the **wheel**, of integers that are relatively prime to all primes of the basis[**?**]. The basis, B, and the wheel, W, are all subsequences of S. Then, we divide n successively by the element in the wheel until we find a divisor. The following theorem and Chinese Reminder Theorem helps us generate W.

Theorem 3.2. *Suppose p is a prime, if $pm|n$ for some $m \in \mathbb{N}$ and $m \geq 1$, then $p|n$.*

If you want to learn more about this article, please contact ysu24@colby.edu.