Application of Prime Numbers in Computer Science and the Algorithms Used To Test the Primality of a Number

Tejash Desai

Veermata Jijabai Technological Institute, H.R. Mahajani Marg, Matunga Road, Mumbai, Maharashtra, India – 400019

Abstract: ***This paper throws light on some of the important properties exhibited by a prime number. It also attempts to list down the fields in which prime numbers are especially important and the reasons behind their importance. Additionally, it gives a detailed explanation of some of the algorithms used to check if a given number is prime and/or to list the prime numbers between two given numbers.***

Keywords: prime numbers, algorithms, mathematics, number theory

1. Introduction

Every integer greater than 1 is classified as either prime or composite. This classification forms the basis of the number theory we know today. This paper attempts to present an overview of prime numbers, their properties and their importance in various fields of study. It also attempts to throw light on some attempts at finding out the primality of a number i.e. to check if a given number is prime. Various algorithms have been presented in history, however none of them is satisfactorily fast or efficient. In fact, the very difficulty involved in finding the primality of a number along with their peculiar properties is the reason why prime numbers have such wide applications in today‟s world.

2. Definition of Prime Number

Definition 1 (English): A prime number (or a prime) is a natural number which has exactly two distinct natural number divisors.

Definition 2 (Mathematical): A natural number which is greater than 1 is a prime number (or a prime) if the following condition is satisfied:

$$\forall\, b \in N \; b|a \Rightarrow b = 1 \vee b = a. \qquad (1)$$

Conversely, a **composite number** may be defined as a natural number which can be represented as the product of 2 natural numbers, none of which is itself.

By the above definitions, 1 is considered to be neither prime nor composite. Also, the discussion of prime and composite numbers is restricted to positive integers only.

3. Importance of Prime Numbers

3.1 Number Theory

Any integer greater than 1 is either a prime or a product of primes. This can be proved easily for all integers via induction. This means that we can define any integer greater than 1 as product of one or more elements from the set of all prime numbers. Conversely, a combination of prime numbers can be multiplied to produce any number at all.

3.2 Cryptography

3.2.1) The **RSA system** in cryptography uses prime numbers widely to calculate the public and the private keys. The strength of this system relies up on the difficulty of factoring large numbers - specifically the difficulty associated with the finding of the specific pair of prime numbers selected to create a large integer called the modulus.

3.2.2) **Diffie-Hellman Key Exchange** in cryptography uses prime numbers in a similar way. It uses a large prime number p as a common modulus based on which two entities, say A and B can communicate securely using their private, undisclosed keys. It is mainly based on the property that if both A and B choose a private key, say „a‟ and „b‟ respectively, and agree upon a number, say, „g‟ publicly, where „g‟ is less than „p‟, then both A and B can send a message to the other one as follows:

A‟s message = $M_1 = g^a$ modulo p

B‟s message = $M_2 = g^b$ modulo p

Then,

$X = M_2^a$ modulo p = M_1^b modulo p = $g^{(a*b)}$ modulo p

Is the message shared. Its security lies in the difficulty involved in finding the shared message without knowing either of the private keys.

3.3 Gödel's numbering

Gödel numbering is a function that assigns to each expression, a unique natural number called its Gödel number. Kurt Gödel, the creator used prime numbers to encode every number in the sequence. Since a prime number has no smaller prime factors, every expression can have only one Gödel number, removing ambiguity. Also, every Gödel number can be mapped to only one expression. Moreover, we can use this function to determine whether a given number is Gödel number or not.[2]

3.4 Computer Science - Calculating Hash Codes

A hash code is a number code for every object that is created by a program. Hash codes are required for quick retrieval/storing of complex objects from/in a hash table. Hash codes need to be reasonably unique for each object so that correctness is maintained. Prime numbers are used in computing hash codes for this reason. For example, Java calculates hash codes for strings as follows:

$$s[0] * 31^{(n-1)} + s[1] * 31^{(n-2)} + \ldots + s[n-1] * 31^{0}$$

where „s" is a string of „n" characters numbered from 0 - n-1 and s[x] signifies the ASCII value of the xth character in s.

The number 31 is a prime number that is close to a power of 2 (It is actually a Mersenne prime number as we shall see later). Prime numbers are chosen because they best distribute data across the hash buckets. Since they have no factors except for themselves and 1, the function “a modulo x” is guaranteed to produce a wider range of answers if x is prime than if it‟s not and thus the number of hash buckets increase.

4. Properties of Prime Numbers

4.1 Mersenne Primes

Mersenne Primes are primes that are 1 less than a power of 2. They can be expressed as $M_n = 2^n$-1. It can be shown that if 2^n-1 is prime, then n is prime as well. The converse, however, is not true. For example, 11 is a prime, but $M_{11} = 2^{11}$-1 = 2047 is not. The largest known prime till today is a Mersenne Prime and has a value of $2^{57,885,161}$-1. [3] Because in binary system, an n-digit number can hold up to 2^n-1 digits, Mersenne primes can be represented efficiently in binary system without needing any extra space. **The Mersenne Twister** is a good pseudorandom number generating algorithm developed by Makoto Matsumoto and Takuji Nishimura.

4.2 Perfect Numbers

A positive integer n is called a perfect number if it is equal to the sum of all of its positive divisors, excluding itself. It can be shown that an even integer greater than 1 is perfect if and only if it has the form $2^{(n-1)} * (2^n - 1)$ and the latter term, 2^n - 1, is prime, (In fact, it is a Mersenne Prime) i.e. any perfect number is a Mersenne prime multiplied by some power of 2.[1] For example, $6 = 2 + 3 + 1 = 2^1 * (2^2\text{-}1)$ is a perfect number. 28 is the next known perfect number after 6. Perfect numbers exhibit the interesting property that the log of such numbers is equal to the sum of the log of their factors (excluding themselves). For example,

$$\log(6) = \log(2 * 3 * 1) = \log(2) + \log(3) + \log(1)$$

4.3 Goldbach’s Conjecture

In his famous letter to Leonhard Euler dated 7 June 1742, Christian Goldbach first conjectures that “every number that is a sum of two primes can be written as a sum of as many primes as one wants.”[4] This is equivalent to saying that "every even number is a sum of two primes". Because Goldbach considered 1 as prime, we can rephrase the conjecture as "every even number greater than 2 is a sum of two primes”. The Goldbach conjecture has been verified and found to be true for all numbers up to $4*10^{14}$. [4] However, whether it is true for all numbers is still not known.

4.4 Relatively Prime Numbers

Two integers are said to be relatively prime if they have no common factors other than 1. In formal notation, this is expressed as:

$$\gcd(M, N) = 1$$

For example, 7 and 8 are relatively prime, but 10 and 8 are not. 1 is considered to be relatively prime to every number because for any integer „x", gcd(x, 1) = 1.

Theorem 1: The least integer with which a prime number is not relatively prime is itself.

Proof: We can prove this theorem easily by considering the definitions of prime numbers and relatively prime pairs. By the definition of a prime number, it divides only 1 and itself. Hence, the only factors that a prime number has are 1 and itself. Thus, it will be relatively prime with every integer between 1 and itself (exclusive). Because every number divides itself, it cannot be relatively prime to itself. Hence, we prove the above theorem

.

4.5 Euler’s Totient Function

Euler‟s totient function (sometimes called Euler‟s *phi* function), written φ(n), returns the number of integers less than *n* and relatively prime to n. For example, φ(8) = 4 {1,2,4,6}. By Theorem 1, the value of the Euler‟s Totient function for any prime number p is p-1. Also, because every number divides itself, p-1 is the maximum value Euler‟s Totient function can have for any integer p. This is also the reason why prime numbers are used to calculate hash-codes – because they are co-prime with the maximum number of integers.

5. Primality of a Number

A correct and fast algorithm to check whether a number is prime has been seeked for a long time by mathematicians and computer scientists. Below are a few known algorithms for checking the primality of a number.

5.1 Sieve of Eratosthenes

This is perhaps the most widely accepted algorithm which has been used since ancient times. This algorithm lists out all the numbers from 1 to n (or 2 to n, since 1 is known to be neither prime nor composite) and then marks off all the multiples of the first prime, which is 2. After marking off all the multiples of 2, it then begins to mark off all the multiples of the next prime, which is 3. It does this routine till the time there are no primes less than n. In the end, all the unmarked numbers are all the primes less than n.

Theorem 2: If we have marked off all the multiples of all the prime numbers up to and including p, then the least multiple of the next prime number p + x (where x is even) that is unmarked is $(p + x)^2$.

Proof By Induction: Let L = {2, 3, 5, …, p, (p + x)} be the set of consecutive prime numbers less than p+x+1. Our aim is to mark off from an infinite set of natural numbers greater than 2, all the multiples of each prime, starting from the first element in the list i.e. 2. Thus after the consideration of 2, we have the following loop invariant which holds true.

$$\forall\, b \in N\ \{\text{markedOff}(2 * b)\}$$

Where markedOff(m) is a predicate denoting that „m‟ is marked off as composite

We then consider 3 and mark off all the multiples of 3 in addition to the ones that are already marked off during the consideration of 2. Thus, we maintain the above loop invariant as

$$\forall\, a \in \{2, 3\}\ \forall\, b \in N\ \{\text{markedOff}(a * b)\}$$

By induction, we state the above loop invariant after considering all the primes up to and including p and marking off their multiples.

$$\forall\, a \in \{2, 3, 5, \ldots, p\}\ \forall\, b \in N\ \{\text{markedOff}(a * b)\}$$

Where {2, 3, 5, …, p} = L – (p + x)

If the above loop invariant holds true and since a*b = b*a, all the elements belonging to {(p + x)*2, (p + x)*3, (p + x)*4 = (p + x)*2*2, (p + x)*5,….(p + x)*(p + x - 1)} are already marked off. Hence the least multiple of (p + x) that is unmarked is (p + x)*(p + x) i.e. $(p + x)^2$.

This theorem will allow us to limit our iterations up to √n, thus increasing the efficiency of the algorithm.

Algorithm Sieve of Eratosthenes (Input: n)

```
Let A be a list of integers from 2 to n (inclusive);
For i = 2, 3, 4,... up to n:
  If A[i] is not marked off:
    For j = $i^2$; $i^2 + i$, $i^2 + 2i$, $i^2 + 3i$,… not exceeding n:
      Mark off A[j];

Output: all unmarked A[i];
```

5.2 Sieve of Atkin

Sieve of Atkin is an improvement over Sieve of Eratosthenes. It maintains a sieve list of all the numbers up to the number in consideration. Each entry in the list corresponds to a mark denoting if that number is prime or not. It should be noted that ~A[i] operation flips the value of A[i] i.e. marks A[i] if it was previously unmarked, and vice versa.

Algorithm Sieve of Atkin (Input: n)

```
1) Initialize the sieve list A with every entry less than n and
marked as non-prime;
2) For every entry in the sieve list A[i] with the modulo
60 equal to r:
If r ∈ {1, 13, 17, 29, 37, 41, 49, 53}:
  For each possible solution to $4x^2 + y^2 = n$:
    A[i] = ~A[i];
Else If r ∈ {7, 19, 31, 43}:
  For each possible solution to $3x^2 + y^2 = n$:
    A[i] = ~A[i];
Else If r ∈ {11, 23, 47, 59}:
  For each possible solution to $3x^2 - y^2 = n$ when x > y:
    A[i] = ~A[i];
Else :
  Ignore;
3) Add 2, 3 and 5 to the result list;
4) Include the next number marked prime in the sieve list
greater than the last number in result list;
5) Mark all the multiples of the square of that number as non-
prime;
6) Repeat steps 4-5 till we get all the prime numbers up to n;
7) Output result list;
```

5.3 Using Fermat's Little Theorem

If *p* is a prime which does not divide the integer *a*, then:

$$a^{p-1} = 1 \pmod{p} \tag{2}$$

This is called Fermat‟s Little Theorem. It can be also stated as:

$$a^{p} = a \pmod{p} \tag{3}$$

It should be noted that every iteration of the following algorithm only increases the confidence in the primality of a given number. In order to be 100% sure, we need to test the above theorem for every value of „a‟ less than „p‟. However, in many cases, absolute confidence is not required, rather, only a sufficient amount of confidence is needed.

Algorithm Primality Check by Fermat's Little Theorem (Input: n)

```
If enough confidence in primality of n:
  Output "Prime";
Let x be any random number less than n;
If ($x^n$ modulo n) not equal to x:
  Output "Not Prime";
Else:
  Increase confidence in n and repeat for a different value
  of x;
```

Fermat‟s Little Theorem holds true for every prime number. However, there are some composite numbers that are detected as prime using Fermat's Little Theorem because they satisfy Fermat‟s Little Theorem. These numbers are called Carmichael numbers. It would be useful to know the list of such numbers beforehand so that we can address them appropriately if we are using the above algorithm.

5.4 Rabin-Miller Test

Rabin-Miller test is an improvement over Fermat's Test in the sense that it handles Carmichael numbers as well. It uses the property that any multiple of 2 can be represented as $2^x * y$ where x and y are some positive integers. For example, $2 = 2^1 * 1$, $80 = 2^4 * 5$. If n is prime, then n has to be odd and n-1 has to be even. So n-1 has to satisfy the above property. Additionally, if n is to be prime, y in the above equation has to be prime.

Rabin-Miller Test for Primality (Input: n)

```
If enough confidence on primality of „n‟:
    Output "Prime";
Express (n – 1) as $2^x * y$;
Pick any random number „a‟ between 2 and n-1, exclusive;
If ($a^y$ modulo n) = 1 or -1:
    Increase confidence in „n‟ and repeat for a different value
    of „a‟;
Else:
    Output "Not Prime";
```

6. Conclusion

This paper, thus consisted of a brief summary of what prime numbers are and what are they used for. It also presented some well-known algorithms for testing the primality of a number. It should be noted that recent algorithms and papers have been written which claim to test the primality in much faster ways. “Primes is in P” is a paper written by Manindra Agrawal, Neeraj Kayal and Nitin Saxena of IIT Kanpur, India in which they have discussed a method for checking the primality of a number in linear time. “An Introduction to Prime Number Sieves” by Jonathan Sorenson, University of Wisconsin-Masison is another paper discussing the various sieves and algorithms running in linear and sublinear time.

References

[1] Jerry Crow, “Prime Numbers in Public Key Cryptography – An Introduction”, SANS Institute Reading Room

[2] “Lecture 23 - Unsolvable Problems in Logic”, Cornell University, CS 4860 Spring, 2009

[3] GIMPS (Greatest Internet Mersenne Prime Search), Orlando, Florida [Online]

[4] Jörg Richstein, “Verifying The Goldbach Conjecture Upto 4.10^4”, Mathematics of Computation, Volume 70, Number 236, Pages 1745-1749