Primality Testing- Is Randomization worth Practicing?

Shubham Sahai Srivastava

Indian Institute of Technology, Kanpur

ssahai@cse.iitk.ac.in

April 5, 2014

Overview

Primes : The fundamental building blocks of a number.

Prime Number

A prime number (or a prime) is a natural number greater than 1 that has no positive divisors other than 1 and itself.

Example : 2, 3, 5, 7, 11, 13 .....

Primes : The fundamental building blocks of a number.

Prime Number

A prime number (or a prime) is a natural number greater than 1 that has no positive divisors other than 1 and itself.

Example : 2, 3, 5, 7, 11, 13 .....

Composite Number

A natural number greater than 1 that is not a prime number is called a composite number.

Example : 4, 6, 8, 10, 12, 15 .....

Carl Friedrich Gauss

"The problem of distinguishing prime numbers from composites, and of resolving composite numbers into their prime factors, is one of the most important and useful in all of arithmetic. . . . The dignity of science seems to demand that every aid to the solution of such an elegant and celebrated problem be zealously cultivated."

Primes : The fundamental building blocks of a number.

Fundamental Theorem of Arithematic

Every integer greater than 1, either is prime itself or is the product of prime numbers.

Also, although the order of the primes in the second case is arbitrary, the primes themselves are not.

Example :

- $330 = 2 \times 3 \times 5 \times 11$
- $1200 = 2^4 \times 3^1 \times 5^2 = 3 \times 2 \times 2 \times 2 \times 2 \times 5 \times 5 = \cdots etc.$

Some Interesting Points

- **Euclid's Theorem :** There are infinitely many prime numbers.
- **Goldbach Conjecture :** Every even number greater than 2 can be written as a sum of two primes.
- **Twin Prime Conjecture :** There are infinitely many primes p such that p + 2 is also prime.
- **Prime Number Theorem :** Number of primes $\leq x \approx \frac{x}{\log_e x}$

Primality Testing

PRIMES

$$PRIMES = \{bin(n) | n \geq 2 \text{ is a prime number}\}$$

SO, Primality Testing algorithm is any algorithm which decides that given any input n, whether $bin(n) \in PRIMES$?

Primality Testing

PRIMES

$$PRIMES = \{bin(n) | n \geq 2 \text{ is a prime number}\}$$

SO, Primality Testing algorithm is any algorithm which decides that given any input n, whether $bin(n) \in PRIMES$?

Which Complexity Class contains PRIMES ?

Primality Testing

PRIMES

$$PRIMES = \{bin(n) | n \geq 2 \text{ is a prime number}\}$$

SO, Primality Testing algorithm is any algorithm which decides that given any input n, whether $bin(n) \in PRIMES$?

Which Complexity Class contains PRIMES ?

Examples :

- Trial Division Test
- Fermat's Test based Primality test
- Miller-Rabin primality test
- Solovay-Strassen primality test
- AKS primality test

Trial Division Test

Algorithm 1 : Trial Division Test

Require: Integer $n \geq 2$

```
1: i : integer
2: i ← 2
3: while i.i ≤ n do
4:     if i divides n then
5:         return COMPOSITE
6:     end if
7:     i ← i + 1
8: end while
9: return PRIME
```

- This algortithm never gives an error
- The running time of the algorithm is exponential *(In terms of number of binary bits needed to represent the number)*
- Several minor optimizations may be carried out, but not much gain in the time complexity.

Trial Division Test : Is it good enough?

- For moderately large n, this algorithm can be used for a calculation by hand.

Trial Division Test : Is it good enough?

- For moderately large n, this algorithm can be used for a calculation by hand.
- As the value of n grows, a computer may be used to carry out the desired calculations.

Trial Division Test : Is it good enough?

- For moderately large n, this algorithm can be used for a calculation by hand.
- As the value of n grows, a computer may be used to carry out the desired calculations.
- But, what happens when n becomes exceedingly large?

Trial Division Test : Is it good enough?

- For moderately large n, this algorithm can be used for a calculation by hand.
- As the value of n grows, a computer may be used to carry out the desired calculations.
- But, what happens when n becomes exceedingly large?

The following table estimates the usefulness of the Algorithm 1 !

Trial Division Test : Is it good enough?

Number	Decimal Digits	Binary Digits	Running Time
11	2	4	0.069 sec
191	3	8	0.081 sec
7927	4	13	0.111 sec
1300391	7	21	0.34 sec
179426549	9	28	13.56 sec
32416190071	11	35	1 hr 33 min 23.5 sec

Table: Running time vs n

These tests were carried out on a core i5 machine with 8 GB RAM

Trial Division Test : Is it good enough?

A 62 digit giant

74838457648748954900050464578792347604359487509026452654305481

- The 62 digit number above happens to be a prime.
- The loop happens to run for more than 10^{31} rounds.
- Even after applying several tricks and optimizations, and under the assumption that a very fast computer is used that can carry out one trial division in 1 nanosecond, say, a simple estimate shows that this would take more than 10^{13} years of computing time on a single computer.

Trial Division Test : Is it good enough?

A 62 digit giant

74838457648748954900050464578792347604359487509026452654305481

- The 62 digit number above happens to be a prime.
- The loop happens to run for more than 10^{31} rounds.
- Even after applying several tricks and optimizations, and under the assumption that a very fast computer is used that can carry out one trial division in 1 nanosecond, say, a simple estimate shows that this would take more than 10^{13} years of computing time on a single computer.

There are several real world algorithms that make use of prime numbers of this magnitude

Example: RSA System

Lets Explore !!

Stated by Pierre de Fermat in 1640.

Fermat's Little Theorem

If p is a prime number, and $1 \leq a < p$. then $a^{p-1} \equiv 1 \, (mod \ \mathrm{p})$

Lets Explore !!

Stated by Pierre de Fermat in 1640.

Fermat's Little Theorem

If p is a prime number, and $1 \leq a < p$. then $a^{p-1} \equiv 1 \, (mod \ \mathrm{p})$

Points to note :

- All prime numbers will satisfy the above thorem.
- Some composite number *may or may not* satisfy it.
- Any number which does not satisfy the Fermat's Little Theorem, is for sure a composite number.

Lets Explore !!

Stated by Pierre de Fermat in 1640.

Fermat's Little Theorem

If p is a prime number, and $1 \leq a < p$. then $a^{p-1} \equiv 1 \, (mod \ \mathrm{p})$

Points to note :

- All prime numbers will satisfy the above thorem.
- Some composite number *may or may not* satisfy it.
- Any number which does not satisfy the Fermat's Little Theorem, is for sure a composite number.

Can we use these properties to design a Primality Test ?

Fermat's Test

Let us take $a = 2$, and for given n, calculate $f(n) = 2^{n-1}$ mod n.

n	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$f(n)$	1	0	1	2	1	0	4	2	1	8	1	2	4	0	1

Table: a^{n-1} mod n, for $a = 2$

- For prime numbers $n \leq 17$, we get $f(n) = 1$
- For non Primes we get some value different from 1.

Fermat's Test

Let us take $a = 2$, and for given n, calculate $f(n) = 2^{n-1}$ mod n.

n	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$f(n)$	1	0	1	2	1	0	4	2	1	8	1	2	4	0	1

Table: a^{n-1} mod n, for $a = 2$

- For prime numbers $n \leq 17$, we get $f(n) = 1$
- For non Primes we get some value different from 1.
- By Fermat's Little Theorem, if a^{n-1} mod n $\neq 1$ we have a definite certificate for the fact that n is composite.
- We call such a, as **F-Witness** for n.
  (Or, more exactly, witness of the fact that n is composite)

Fermat's Test

Let us take $a = 2$, and for given n, calculate $f(n) = 2^{n-1}$ mod n.

n	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$f(n)$	1	0	1	2	1	0	4	2	1	8	1	2	4	0	1

Table: a^{n-1} mod n, for $a = 2$

- For prime numbers $n \leq 17$, we get $f(n) = 1$
- For non Primes we get some value different from 1.
- By Fermat's Little Theorem, if a^{n-1} mod n $\neq 1$ we have a definite certificate for the fact that n is composite.
- We call such a, as **F-Witness** for n.
  (Or, more exactly, witness of the fact that n is composite)
- If n is a prime number than, a^{n-1} mod n $= 1, \forall a | 1 \leq a \leq$ n-1

Fermat's Test

Algorithm 2 : Fermat's Test

Require: Odd Integer $n \geq 3$

1: $i \leftarrow 0$
2: **repeat**
3: Let a be randomly chosen from $\{2, \cdots, n-2\}$
4: **if** a^{n-1} mod n $\neq 1$ **then**
5: **return** COMPOSITE
6: **end if**
7: $i \leftarrow i + 1$
8: **until** $i < k$
9: **return** PRIME

- If the algorithm outputs COMPOSITE, then n is guaranteed to be composite.
- The running time of the algorithm depends on calculation of a^{n-1} mod n *(which takes $O(\log n)$ arithematic operations.)*
- But, the algorithm might give wrong output !!

Fermat's Test : When will it give error?

When will the algorithm give a wrong output ?

Fermat's Test : When will it give error?

When will the algorithm give a wrong output ?

- If the number is prime the algorithm will always give the output as "PRIME".
- If the input number is composite, the algorithm might claim that the number is prime. [Hence, give an error]

Fermat's Test : When will it give error?

When will the algorithm give a wrong output ?

- If the number is prime the algorithm will always give the output as "PRIME".
- If the input number is composite, the algorithm might claim that the number is prime. [Hence, give an error]

Why is this error generated?

Fermat's Test : When will it give error?

When will the algorithm give a wrong output ?

- If the number is prime the algorithm will always give the output as "PRIME".
- If the input number is composite, the algorithm might claim that the number is prime. [Hence, give an error]

Why is this error generated?

- Due to the presence of F-Liars

F-liar

For an odd composite number n we call an element a, $1 \leq a \leq n-1$, an F-liar if $a^{n-1} \bmod n = 1$

Fermat's Test : Error Probability

When is the probability that the algorithm give a wrong output ?

Let,

- Let $Z_n^* = \{a | 1 \leq a < n, gcd(a, n) = 1\}$
- And the operations defined in Z_n^* be $+_n$ and $\times_n$
- $L^F = \{a | 1 \leq a < n, a^{n-1} \bmod n = 1\}$

Theorem

If $n \geq 3$ is an odd composite number such that there is at least one F-witness a in Z_n^* , then the Fermat test applied to n gives answer 1 with probability more than $\frac{1}{2}$.

Fermat's Test : Error Probability

Theorem

If $n \geq 3$ is an odd composite number such that there is at least one F-witness a in Z_n^* , then the Fermat test applied to n gives answer 1 with probability more than $\frac{1}{2}$.

We know that L^F is a subset of Z_n^*.
Since Z_n^* is a finite group, and
(a) $1 \in L^F, since 1^{n-1} = 1$
(b) L^F is closed under operations in Z_n^*, since
if a^{n-1} mod n $= 1$ and b^{n-1} mod n $= 1$,
then $(ab)^{n-1} \equiv a^{n-1} \cdot b^{n-1} \equiv 1 \cdot 1 \equiv 1 \pmod{n}$

Fermat's Test : Error Probability

Theorem

If $n \geq 3$ is an odd composite number such that there is at least one F-witness a in Z_n^* , then the Fermat test applied to n gives answer 1 with probability more than $\frac{1}{2}$.

We know that L^F is a subset of Z_n^*.
Since Z_n^* is a finite group, and
(a) $1 \in L^F, since 1^{n-1} = 1$
(b) L^F is closed under operations in Z_n^*, since
if a^{n-1} mod n $= 1$ and b^{n-1} mod n $= 1$,
then $(ab)^{n-1} \equiv a^{n-1} \cdot b^{n-1} \equiv 1 \cdot 1 \equiv 1 \pmod{n}$
Hence, L^F is a proper subgroup of Z_n^*
This gives us the bound that $|L^F| \leq (n-2)/2$

Fermat's Test : Error Probability

Theorem

If $n \geq 3$ is an odd composite number such that there is at least one F-witness a in Z_n^* , then the Fermat test applied to n gives answer 1 with probability more than $\frac{1}{2}$.

We know that L^F is a subset of Z_n^*.
Since Z_n^* is a finite group, and
(a) $1 \in L^F, since 1^{n-1} = 1$
(b) L^F is closed under operations in Z_n^*, since
if a^{n-1} mod n $= 1$ and b^{n-1} mod n $= 1$,
then $(ab)^{n-1} \equiv a^{n-1} \cdot b^{n-1} \equiv 1 \cdot 1 \equiv 1$ (mod n)
Hence, L^F is a proper subgroup of Z_n^*
This gives us the bound that $|L^F| \leq (n-2)/2$
Hence, probability that a number randomly chosen from $\{2, \cdots, n-2\}$ in in $L^F < \frac{1}{2}$

Carmichael Numbers

Carmichael Number

An odd composite number n is called a Carmichael number if:
$a^{n-1} \bmod n = 1$, for all $a \in Z_n^*$,

where
$Z_n^* = \{a | 1 \leq a < n, gcd(a, n) = 1\}$

- The smallest Carmichael number is 561.
- In 1994 was it shown that there are infinitely many Carmichael numbers.
- If the Carmichael Number is fed into the Fermat's Test, the probability that a wrong answer PRIME is given is close to 1.

Hence Fermat's test fail for Carmichael Numbers.