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 ?

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 $n \geq 2$

```
i : integer
i ← 2
while i.i ≤ n do
    if i divides n then
        return COMPOSITE
    end if
    i ← i + 1
end while
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 $a^{p-1} \equiv 1 \, (mod \text{ p})$

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 $a^{p-1} \equiv 1\,(mod\ \text{p})$

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.

Stated by Pierre de Fermat in 1640.

Fermat's Little Theorem

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

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 $n \geq 3$

```
i ← 0
repeat
    Let a be randomly chosen from {2, ··· , n − 2}
    if a^(n−1) mod n ≠ 1 then
        return COMPOSITE
    end if
    i ← i + 1
until i < k
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 $a^{n-1} \bmod n = 1$

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 $(ab)^{n-1} \equiv a^{n-1} \cdot b^{n-1} \equiv 1 \cdot 1 \equiv 1$ (mod 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 $(ab)^{n-1} \equiv a^{n-1} \cdot b^{n-1} \equiv 1 \cdot 1 \equiv 1$ (mod 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 $(ab)^{n-1} \equiv a^{n-1} \cdot b^{n-1} \equiv 1 \cdot 1 \equiv 1 \pmod{n}$
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.

$\boxed{\text{Hence Fermat's test fail for Carmichael Numbers.}}$

Nontrivial Square Roots of 1

Let's consider one more property of arithmetic modulo p, which we could use as a certificate of compositeness.

Nontrivial Square Roots of 1

Let's consider one more property of arithmetic modulo p, which we could use as a certificate of compositeness.

Square Roots of 1

Let $1 \leq a < n$. Then a is called a **square root of 1 modulo n** if: $a^2 \bmod n = 1$.

Nontrivial Square Roots of 1

Let's consider one more property of arithmetic modulo p, which we could use as a certificate of compositeness.

Square Roots of 1

Let $1 \leq a < n$. Then a is called a **square root of 1 modulo n** if: $a^2 \bmod n = 1$.

- 1 and n-1 are trivial square roots of 1 modulo n.

Nontrivial Square Roots of 1

Let's consider one more property of arithmetic modulo p, which we could use as a certificate of compositeness.

Square Roots of 1

Let $1 \leq a < n$. Then a is called a **square root of 1 modulo n** if: $a^2 \bmod n = 1$.

- 1 and n-1 are trivial square roots of 1 modulo n.
- If n is a prime number, there are no other square roots of 1 modulo n.

Nontrivial Square Roots of 1

Let's consider one more property of arithmetic modulo p, which we could use as a certificate of compositeness.

Square Roots of 1

Let $1 \leq a < n$. Then a is called a **square root of 1 modulo n** if: $a^2 \bmod n = 1$.

- 1 and n-1 are trivial square roots of 1 modulo n.
- If n is a prime number, there are no other square roots of 1 modulo n.
- Thus, if we find some nontrivial square root of 1 modulo n, then n is certainly composite.

Nontrivial Square Roots of 1

Let's consider one more property of arithmetic modulo p, which we could use as a certificate of compositeness.

Square Roots of 1

Let $1 \leq a < n$. Then a is called a **square root of 1 modulo n** if:
$a^2 \bmod n = 1$.

- 1 and n-1 are trivial square roots of 1 modulo n.
- If n is a prime number, there are no other square roots of 1 modulo n.
- Thus, if we find some nontrivial square root of 1 modulo n, then n is certainly composite.
- More generally, if $n = p_1 \cdot p_2 \cdots p_r$, for distinct odd primes $p_1, p_2 \cdots p_r$, then there are exactly 2^r square roots of 1 modulo n

Nontrivial Square Roots of 1

Let's consider one more property of arithmetic modulo p, which we could use as a certificate of compositeness.

Square Roots of 1

Let $1 \leq a < n$. Then a is called a **square root of 1 modulo n** if: $a^2 \bmod n = 1$.

- 1 and n-1 are trivial square roots of 1 modulo n.
- If n is a prime number, there are no other square roots of 1 modulo n.
- Thus, if we find some nontrivial square root of 1 modulo n, then n is certainly composite.
- More generally, if $n = p_1 \cdot p_2 \cdots p_r$, for distinct odd primes $p_1, p_2 \cdots p_r$, then there are exactly 2^r square roots of 1 modulo n
- This means that unless n has extremely many prime factors, it is useless to try to find nontrivial square roots of 1 modulo n by testing randomly chosen a.

Fermat's Test

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

- As p is odd, $p - 1$ would be even.
- So, $p - 1 = u \cdot 2^k$, for some odd u and $k \geq 1$
- Thus, $a^{p-1} \equiv ((a^u)\text{mod } n)^{2^k}\text{mod n}$
- This means that we may calculate $a^{n-1} mod$ n with k+1 intermediate steps, if we let:
  $b_0 = a^u$ mod n; $b_i = {b^2}_{i-1}$ mod n; for $i = 1 \cdots k$

Example

Let us take $n = 325$. So, $324 = 81 \cdot 2^2$

a	$b_0 = a^{81}$	$b_1 = a^{162}$	$b_2 = a^{324}$
2	252	129	66
7	307	324	1
32	57	324	1
49	324	1	1
65	0	0	0
126	1	1	1
201	226	**51**	1
224	**274**	1	1

Table: a^{n-1} mod n,with intermediate steps for $n = 325$

- 2, 65 are a F-witness for 325.
- 7, 32, 49, 126, 201, 224 are F-liars

Possible Cases

b_0	b_1	$\cdots$				$\cdots$	b_{k-1}	b_k	Case
1	1	$\cdots$	1	1	1	$\cdots$	1	1	No Info.
n-1	1	$\cdots$	1	1	1	$\cdots$	1	1	No Info.
*	*	$\cdots$	*	n-1	1	$\cdots$	1	1	No Info.
*	*	$\cdots$	*	*	*	$\cdots$	*	n-1	Composite
*	*	$\cdots$	*	*	*	$\cdots$	*	*	Composite
*	*	$\cdots$	*	1	1	$\cdots$	1	1	Composite
*	*	$\cdots$	*	*	*	$\cdots$	*	1	Composite

Table: Powers of a^{n-1} mod n,with intermediate steps, possible cases

Miller Rabin Test

Algorithm 3 : Miller Rabin Test

For u odd and k so that $n - 1 = u.2^k$
Let a be randomly chosen from $\{2, \cdots, n-2\}$ and $b \leftarrow a^u$ mod n
if $b \in \{1, n-1\}$ **then**
    return PRIME
end if
repeat
    $b \leftarrow b^2$ mod n
    if $b = n - 1$ **then**
        return PRIME
    end if
    if $b = 1$ **then**
        return COMPOSITE
    end if
until $i < k$
return COMPOSITE

Error Probability : Miller Rabin Test

- If n is not a Carmichael Number, the miller rabin test performs better than Fermat's Test.
- Hence, the probability to give an error is at most $\frac{1}{2}$.

Error Probability : Miller Rabin Test

- If n is not a Carmichael Number, the miller rabin test performs better than Fermat's Test.
- Hence, the probability to give an error is at most $\frac{1}{2}$.

Lets see what happens if n is a Carmichael number

Error Probability : Miller Rabin Test

- Let L_n be set that contains all Miller-Rabin Liars (MR-Liar) of number n.
- Our aim would be now to proof that L_n is a proper subgroup of Z_n^*.

Error Probability : Miller Rabin Test

- Let L_n be set that contains all Miller-Rabin Liars (MR-Liar) of number n.
- Our aim would be now to proof that L_n is a proper subgroup of Z_n^*.
- Let i_0 be the maximal $i \geq 0$ such that there is some MR-Liar a_0 with ${a_0}^{u.2^{i_0}} \bmod n = n-1$.
- Since n is a Carmichael number, ${a_0}^{u.2^k} = {a_0}^{n-1} = 1$ mod n. Hence, $0 \leq i_0 < k$

Now, we define :
$B_n = \{a \mid 0 \leq a < n, a^{u.2^{i_0}}$ mod n $\in \{1, n-1\}\}$, and $L_n =$ Set of all MR-Liars for n

Error Probability : Miller Rabin Test

Now, the basic idea would be to prove that L_n is a proper subgroup of Z_n^*.

Error Probability : Miller Rabin Test

Now, the basic idea would be to prove that L_n is a proper subgroup of Z_n^*.

We will prove it in three steps by showing :

Error Probability : Miller Rabin Test

Now, the basic idea would be to prove that L_n is a proper subgroup of Z_n^*.

We will prove it in three steps by showing :

- $L_n \subseteq B_n$
- B_n is a subgroup of Z_n^*
- $Z_n^* - B_n \neq \phi$

Lets look at them one by one !

1. To show : $L_n \subseteq B_n$

Let a be arbitrary MR-Liar.
Case 1 : If a^u mod n = 1. Then, $a^{u.2^{i_0}}$ mod n = 1 as well, and hence $a \in B_n$
Case 2 : If $a^{u.2^i}$ mod n = n-1, for some i. Then, $0 \leq i \leq i_0$.
Now, if $i = i_0$, we directly have $a \in B_n$.
and, if $i < i_0$, then :
$a^{u.2^i_0}$ mod n $= (a^{u.2^i} \bmod n)^{2^{i_0 - i}}$ mod n $= 1$

Hence, $a \in B_n$

2. To show : B_n is a subgroup of Z_n^*

We know that B_n is a subset of Z_n^*.
Since Z_n^* is a finite group, and
(a) $1 \in B_n, since 1^{u.2^i_0} \bmod n = 1$
(b) B_n is closed under operations in Z_n^*.
Let a, b $\in B_n$
Then, $a^{u.2^{i_0}} \bmod n$, $b^{u.2^{i_0}} \bmod n \in \{1, n-1\}$
Since, 1.1 = 1,
1.(n-1) = (n-1).1 = (n-1), and
(n-1).(n-1) mod n = 1,
we have, $(ab)^{u.2^{i_0}} \bmod n = (a^{u.2^{i_0}} \bmod n).(b^{u.2^{i_0}}) \bmod n \in \{1, n-1\}$
Hence, $(ab)^{u.2^{i_0}} \bmod n \in B_n$

So, B_n is a subgroup of Z_n^*

Error Probability : Miller Rabin Test

3. To show : $Z_n^* - B_n \neq \phi$

- We know that any Carmichael number has atleast 3 different prime factors.
- Hence can be written as $n = n_1.n_2$ for odd numbers n_1 and n_2 which are relatively prime.
- We had, ${a_0}^{u.2^{i_0}} \equiv -1 \pmod{n}$
  Let $a_1 = a_0 \bmod n_1$.
- By CRT, there is a unique number $a \in \{0, ..., n-1\}$, with $a \equiv a_1 \pmod{n_1}$ and $a \equiv 1 \pmod{n_2}$
- Calculating modulo n_1, we have that $a \equiv a_1 \pmod{n_1}$, hence $a^{u.2^{i_0}} \equiv -1 \pmod{n_1}$
- Calculating modulo n_2, we have that $a \equiv 1 \pmod{n_2}$, hence $a^{u.2^{i_0}} \equiv 1^{u.2^{i_0}} \equiv 1 \pmod{n_2}$

3. To show : $Z^* - B_n \neq \phi$ (...Continued)

We have,

$a^{u.2^{i_0}} \equiv -1 \pmod{n_1}, \implies a^{u.2^{i_0}} \not\equiv 1 \pmod{n}$

$a^{u.2^{i_0}} \equiv 1^{u.2^{i_0}} \equiv 1 \pmod{n_2} \implies a^{u.2^{i_0}} \not\equiv -1 \pmod{n}$

- This means $a^{u.2^{i_0}} \pmod{n} \notin \{1, n-1\}$, hence $a \notin L_n$
- Further, $a^{u.2^{i_0+1}} \equiv 1 \pmod{n_1}$, and
  $a^{u.2^{i_0+1}} \equiv 1 \pmod{n_2}$.
- Hence, by CRT, $a^{u.2^{i_0+1}} \equiv 1 \pmod{n}$,
- So, $a \in Z^*$
- Hence, $a \in Z^* - B_n \implies Z^* - B_n \neq \phi$

Error Probability : Miller Rabin Test

By the 3 parts above, we can conclude :
B_n **is a proper subgroup of** Z^*

- Hence, $|B_n|$ divides $|Z^*|$
- Also, $|B_n| \neq |Z^*|$
- Therefore, $|B_n| \leq \frac{n}{2}$

By the 3 parts above, we can conclude :
B_n **is a proper subgroup of** Z^*

- Hence, $|B_n|$ divides $|Z^*|$
- Also, $|B_n| \neq |Z^*|$
- Therefore, $|B_n| \leq \frac{n}{2}$

Error Probability : Miller Rabin Test

The error probability of Miller Rabin is $\frac{1}{2}$, for one iteration.

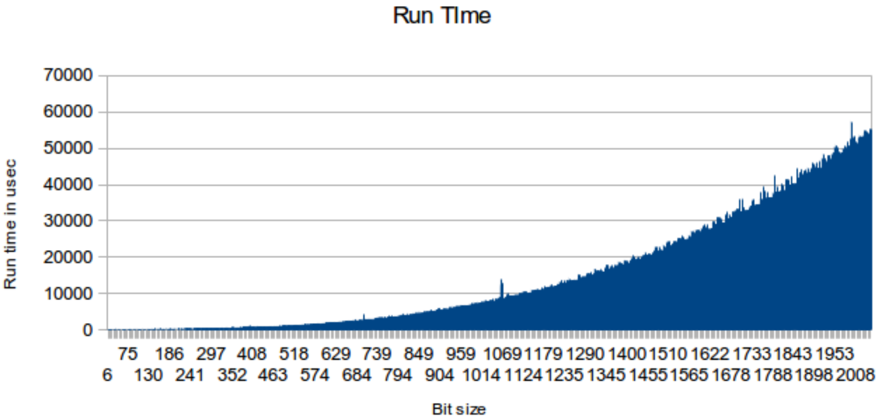
For k iterations of Miller Rabin Test, the probability of error is bounded by $(\frac{1}{2})^k$

Experimental Results

Running Time vs Size of input

- To carry out this analysis, we randomly selected 1000 integers each for bitsize ranging from 2 to 2048.
- Hence, $1000 \times 2047 = 2,047,000$ numbers in total.
- Then the running time was aggregated corresponding to number of bits.
- The result is summarized in the following plot.

Running Time vs Size of input



Dataset Used

- To carry out further analysis, we used the dataset provided by : Center for Experimental and Constructive Mathematics, Simon Fraser University, British Columbia, Canada.
- The dataset was last updated on 25-April-2013.
- It contains data on all base-2 Fermat pseudoprimes below 2^{64}.

Pseudoprimes	Strong Pseudoprimes	Carmichael Numbers
118,968,378	31,894,014	4,279,356

Table: Data Set Statistics

Error Probability

- To analyze the error probability we used the dataset mentioned.
- As we know that all the numbers in the dataset are composites, we recorded the number of primes detected by our algorithm.
- We recorded these number of false positives for differnet number of iterations of the algorithm.
- We expected that, as the number of iteration will increase, the number of false positive will decrease drastically. *(Error Probability* $\leq \frac{1}{2^k}$*)*
- We carried out the experiment for the entire datset, as well as for Carmichael numbers explicitely.
- Our findings are present in the folllowing slides.

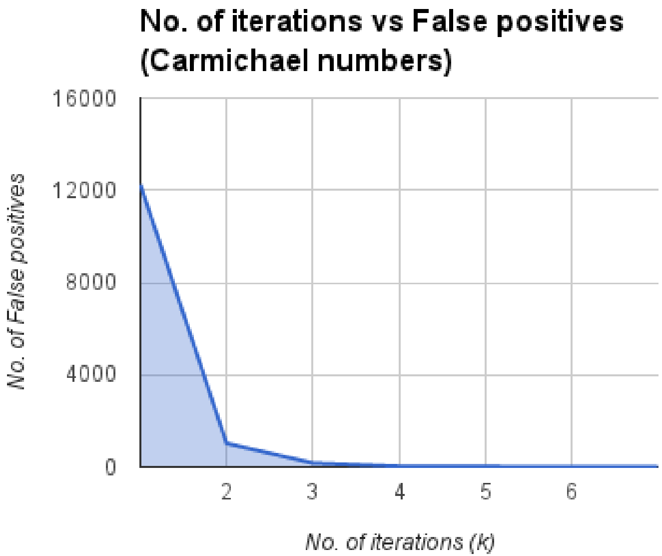
Error Probability (Carmichael Numbers)

The following table summarizes the result of running k iterations of Miller Rabin test on Carmichael Numbers.

Iterations (k)	Number of Composites	Number of primes
1	4267107	12249
2	4278338	1018
3	4279188	168
4	4279328	28
5	4279344	12
6	4279355	1
7	4279356	0

Table: Experimental Result for Carmichael Numbers vs k

Error Probability (Carmichael Numbers)



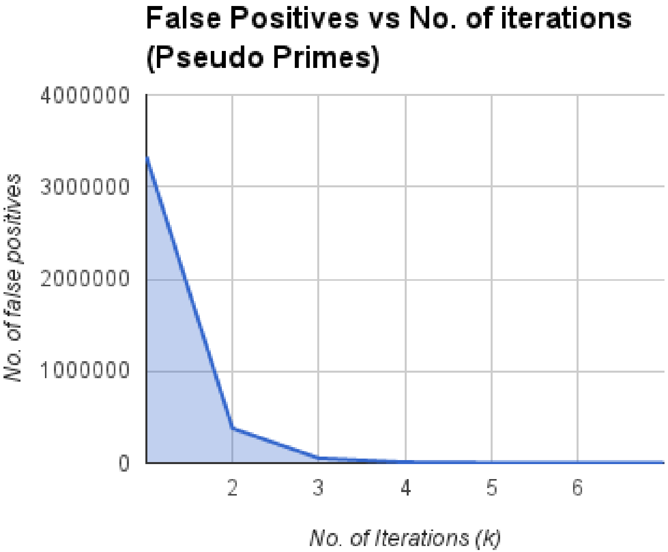
Error Probability (Entire Dataset)

The following table summarizes the result of running k iterations of Miller Rabin test on Entire Dataset.

Iterations (k)	Number of Composites	Number of primes
1	115639122	3329256
2	118592423	375955
3	118915714	52664
4	118960099	8279
5	118967046	1332
6	118968151	227
7	118968331	47
8	118968376	2

Table: Experimental Result for Entire Dataset vs k

Error Probability (Entire Dataset)



Conclusion (Error Probability)

- The Miller Rabin test performs indifferently for Carmichael Numbers (unlike Fermat's Test)
- The number of false positives detected reduces drastically as number of iterations increases.
- For 8 iterations of Miller Rabin, the error reduces to almost 0.

Density of Primes

- For this test, we chose 10^9 integers, randomly, of bit length 64, 128, 256, 512 and 1024.
- We used 5 iterations of Miller Rabin Test, to calculate the number of primes in the set.
- $\mathrm{D} = \frac{\texttt{No.of Primes}}{\texttt{No.of sample numbers}(=10^9)}$
- The density of primes is given by : $\frac{1}{ln\ t}$
- The following table shows the results.

Density of Primes

The following table compares the value of density of primes that we get (D) with the expected value of density (Density)

Bit Length	**Number of Primes**	**D**	**Density**
64	23164312	.023164	.022542
128	12091211	.012091	.011271
256	5678645	.005678	.00563552
512	2820804	.002820	.0028177
1024	1408923	.001408	.0014088

Table: Density of primes

Divisibility with small prime set

Primes	Least No. that is false positive
2	$341\ (11 \times 31)$
3	$91\ (7 \times 13)$
5	$217\ (7 \times 31)$
7	$25\ (5 \times 5)$
$2, 3$	$1105\ (5 \times 13 \times 17)$
$2, 5$	$561(3 \times 11 \times 17)$
$2, 7$	$561(3 \times 11 \times 17)$
$3, 5$	$1541(23 \times 67)$
$3, 7$	$703(19 \times 37)$
$5, 7$	$561(3 \times 11 \times 17)$
$2, 3, 5$	$1729(7 \times 13 \times 19)$
$2, 3, 7$	$1105(5 \times 13 \times 17)$
$3, 5, 7$	$29341(13 \times 37 \times 61)$

Table: Least Composite that base fails to identify

Conclusion

- Miller Rabin Test, perform equivalently well than any deterministic counterparts.
- It is much more easier to implement compared to deterministic counterpart.
- Miller Rabin is robust enough that it is defacto for working with primes in RSA

Is Randomization worth practicing?

- These randomized algorithms, are sufficient for solving the primality problem for quite large inputs for all practical purposes.

Is Randomization worth practicing?

- These randomized algorithms, are sufficient for solving the primality problem for quite large inputs for all practical purposes.
- For practical purposes, there is no reason to worry about the risk of giving output PRIME on a composite input n.

Is Randomization worth practicing?

- These randomized algorithms, are sufficient for solving the primality problem for quite large inputs for all practical purposes.
- For practical purposes, there is no reason to worry about the risk of giving output PRIME on a composite input n.
- Such a small error probability is negligible in relation to other (hardware or software) error risks that are inevitable with real computer systems.

Is Randomization worth practicing?

- These randomized algorithms, are sufficient for solving the primality problem for quite large inputs for all practical purposes.
- For practical purposes, there is no reason to worry about the risk of giving output PRIME on a composite input n.
- Such a small error probability is negligible in relation to other (hardware or software) error risks that are inevitable with real computer systems.
- Still, from a theoretical point of view, the question remained whether there was an absolutely error-free algorithm for solving the primality problem with a small time bound.

The End