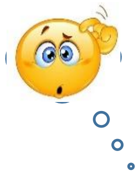
Chapter 00

Introducing Foundations

Contents



Elementary Number-Theoretic Notions

An application of number-theoretic algorithms is in ***cryptography***

- the discipline concerned with encrypting a message sent from one party to another, such that someone who intercepts the message will not be able to decode it.

Let the set Z = { …., -2, -1, 0, 1, 2, 3, ….} of integers.

Let the set N = {0, 1, 2, 3, ….} of natural numbers (nonnegative integers.

The notation $d \mid a$ (read "d ***divides*** a") means

- that $a = k*d$ for some integer k, (i.e., a is k multiple of d).

Congruence Modulo n : m and k are congruent modulo n

Definition of Congruency Modulo n:

Let m and k be integers and n be a positive integer ($n > 0$).

m *is congruent to* k *modulo* n, denoted as

$$m \equiv k \bmod n$$

if, and only if $n \mid (m - k)$.

Or, we said that *m and k are equivalent* ($\equiv$) *mod n*.

Symbolically, $n \mid (m - k) \leftrightarrow m \equiv k \bmod n$.

r = x mod y.

x = q*y + r $\Longrightarrow$ $[\,r\,]_y = \{\, r + q*y \mid q \in Z \,\}$

Let Z be the set of integers {…, -2, -1, 0, 1, 2, … }.

All integers can be partitioned into n equivalence classes, according to their remainders modulo n.

Define the equivalence class modulo n containing an integer a to be

$$[a]_n = \{a + k\,n \mid k \,\varepsilon\, Z\},$$

For example, $[3]_7 = \{ \ldots, -25, -18, -11, -4, 3, 10, 17, 24, 31, 38, \ldots\}$.

i.e., $b \in [a]_n$ iff $b \equiv a \pmod{n}$.

iff $n \mid (b - a)$. i.e., b must be equal to a + kn.

Example 0.47:

Since $5 \mid (33 - 33)$, $33 \equiv 33 \bmod 5$.

Since $5 \mid (33 - 28)$, $33 \equiv 28 \bmod 5$.

Since $5 \mid (33 - 23)$, $33 \equiv 23 \bmod 5$.

Since $5 \mid (33 - 18)$, $33 \equiv 18 \bmod 5$.

Since $5 \mid (33 - 13)$, $33 \equiv 13 \bmod 5$.

Since $5 \mid (33 - 8)$, $33 \equiv 8 \bmod 5$.

Since $5 \mid (33 - 3)$, $33 \equiv 3 \bmod 5$.

Since $5 \mid (33 - (-2))$, $33 \equiv -2 \bmod 5$.

Since $5 \mid (33 - (-7))$, $33 \equiv -7 \bmod 5$. $\boxed{5 \mid (33 - (-7)) \text{ or } 5 \mid (-7-33)}$

$[3]_5 = \{ \ldots, -7, -2, 3, 8, 13, 18, 23, 28, 33, \ldots \}$ is *the equivalence class modulo 5 containing 3.*

Theorem 0.1.4.1 Modular Equivalences

Let a and b and n be any integers and suppose $n > 1$.

The following statements are all equivalent:

1. $n \mid (a - b)$
2. $a \equiv b \pmod{n}$
3. $a = b + kn$ for some integer k
4. a and b have the same (nonnegative) remainder when divided by n
5. $a \bmod n = b \bmod n$.

Proof: Obvious. Example: $5 \mid (33 - 18)$.

Primality testing

Primality testing

Do we have any way to know a number is prime without actually trying to factor the number?

- Fermat's little theorem states that *if* p *is a prime number, then for any integer* a, *the number* $a^p - a$ *is an integer multiple of* p.
- In the notation of modular arithmetic, $a^p \equiv a \pmod{p}$.
- That is, $p \mid (a^p - a)$

i.e., $(a^p - a) \bmod p = 0$.

For example,

- if $a = 2$ and $p = 11$, $2^{11} = 2048$, and
  $2048 - 2 = 186 \times 11$, an integer 186 multiple of 11.
- That is, $11 \mid 2^{11} - 2$, where 11 is a prime. i.e., $2^{11} \equiv 2 \pmod{p}$.
- If $a = 2$, and $p = 12$, $2^{12} = 4096$, then $12 \nmid 2^{12} - 2$

Primality testing

Do we have any way to know a number is prime without actually trying to factor the number?

Fermat's little theorem states that:

- *if* p *is a prime number,*
  then for any integer a, *the number* $a^p - a$ *is an integer multiple of* p.
- In the notation of modular arithmetic, $a^p \equiv a \pmod{p}$.

i.e., $(a^p - a) \bmod p = 0$.
$a(a^{p-1} - 1) \bmod p = 0$
$(a^{p-1} - 1) \bmod p = 0$
$a^{p-1} \equiv 1 \pmod{p}$

If a is not divisible by p, Fermat's little theorem is equivalent to *the statement that* $a^{p-1} - 1$ *is an integer multiple of* p, or in symbols.

$$a^{p-1} \equiv 1 \pmod{p}.$$

$$\gcd(a^{p-1}, p) = 1$$

For $p \mid (a^p - a) = p \mid a(a^{p-1} - 1)$,
if $p \nmid a$, then $p \mid (a^{p-1} - 1)$.

Let formally state:

Fermat's little theorem (1640):

- If p is prime, then for every integer $1 \leq a < p$,

  $a^{p-1} \equiv 1 \pmod{p}$.

- Since $m \equiv k \bmod n$ if, and only if $n \mid (m - k)$,

  $a^{p-1} \equiv 1 \pmod{p}$ if, and only if $p \mid (a^{p-1} - 1)$.
- a^{p-1} *is congruent to* 1 *modulo* n
- $1 \leq a < p$ condition is to define the equivalence classes modulo p. such as $[1]_7$, $[2]_7$, $[3]_7$, $[4]_7$, $[5]_7$, and $[6]_7$.
- If $a = 0$, a^{p-1} is undefined.
- If $a \geq p$ then it will repeat the equivalence classes.

Definition: $m \equiv k \bmod n$ if, and only if $n \mid (m - k)$.

According to Theorem 0.1.4.1 Modular Equivalences, we have

$a^{p-1} \equiv 1 \pmod{p}$ if, and only if $p \mid (a^{p-1} - 1)$.

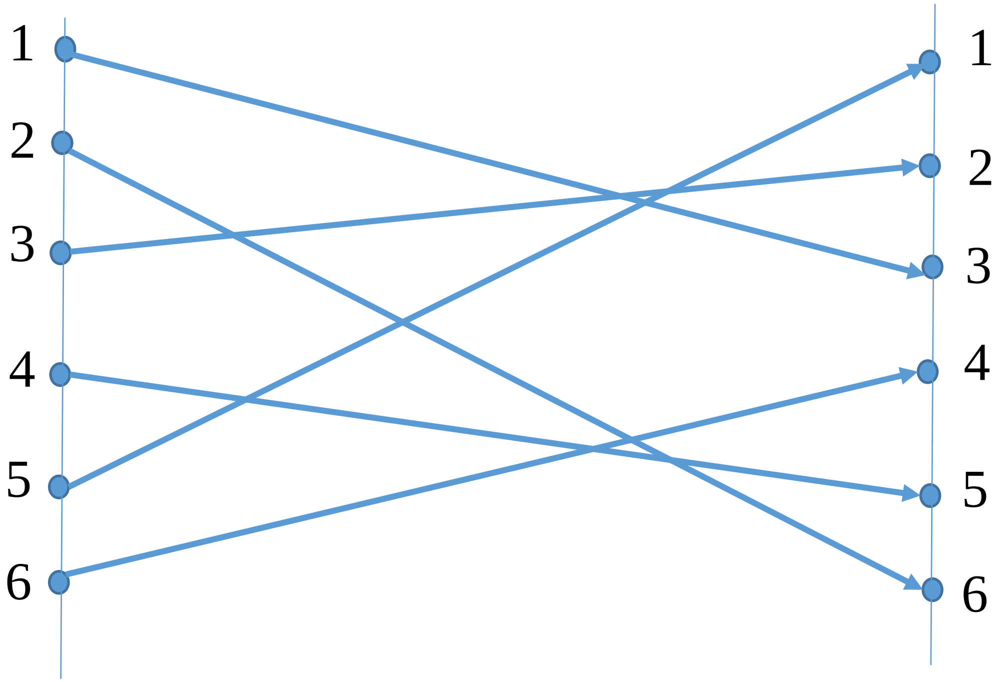
$a^{p-1} = 1 + kp$ for some integer k. Example: $2^{7-1} = 1 + 9*7$

a^{p-1} and 1 have the same (nonnegative) remainder when divided by p

$a^{p-1} \bmod p = 1 \bmod p$.

s * 3 mod 7

s ∈ S   s * a mod p   R (remainder)



Let a = 3, p = 7, 3%7= 3:

If s is 1, 2, 3, 4, 5, 6, then 1*3% 7 = 3; 2*3% 7 = 6; 3*3% 7 = 2;
4*3% 7 = 5; 5*3% 7 = 1; 6*3% 7 = 4;
If s is 8, 9, 10, 11, 12, 13, i.e., 1+1*7=8; 2+1*7= 9;...; 6+1*7= 13;
then 8*3%7= 3; 9*3%7=3;...; 13*3%7=4
... general term of s: s = s+i*p, 0 < i < p

6! mod 7 = 720 mod 7 = 6.

Let p be 7and **$1 \le a < p$.**

$a^{p-1} \equiv 1 \pmod{p}$,

$3^6 * 6! \pmod{7}$

$= 729 * 720 \pmod{7}$

$= 524880 \pmod{7} = 6.$

Or

720 =120* 2*3 %7=6

$729 * 720 \pmod{7}$

$= 729 \pmod{7} * 720 \pmod{7} \pmod{7}$

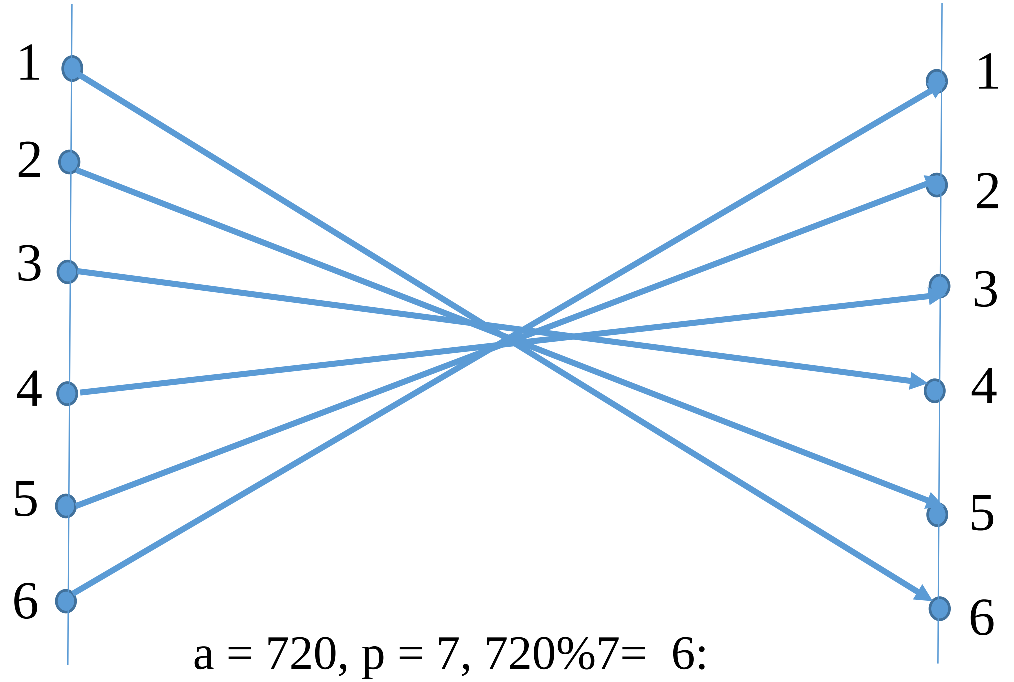
$= 1 * 6 \pmod{7} = 6.$ 729/7 =1

Therefore, $6!$ and $3^6 * 6!$ are of the same class, denoted as

$6! \equiv 3^6 * 6! \pmod{7}$.

s * 720 mod 7

s ∈ S   s * a mod p   R (remainder)



a = 720, p = 7, 720%7= 6:

s is 1, 2, 3, 4, 5, 6: 1*720%7 = 6; 2*720%7 = 5; 3*720%7 = 4;
4*720%7 = 3; 5*720%7 = 2; 6*720%7 = 1;
8, 9, 10, 11, 12, 13: 1+1*7=8; 2+1*7= 9; …; 6+1*7= 13;
15, 16, 17, 18, 19, 20: 1+2*7=15; 2+2*7=16;…, 6+2*7=20
…
729, … : **1**+104*7=729;
Then 729*720%7 ≡ 1*6%7
general term of s: s = s+i*p, 0 < s < p; 0 < i.

6! mod 7 = 720 mod 7 = 6.

Let p be 7and **1 ≤ a < p.**

$a^{p-1} \equiv 1 \pmod{p}$,

$3^6 * 6! \pmod{7}$ [note that $3^6 \equiv 1 \bmod 7$]

= 729 * 720 (mod 7) [1 * 720 mod 7 = 6]

= 524880(mod 7) = 6. [otherwise, do *]

Or

729 * 720 (mod 7)

= 729 (mod 7) * 720 (mod 7)(mod 7)

= 1 * 6 (mod 7) = 6.

Therefore, 6! and 3^6 * 6! are of the same class, denoted as

$6! \equiv 3^6 * 6! \pmod{7}$.

Recall:

Fermat's little theorem (1640):

If p is prime, then for every integer $1 \leq a < p$,

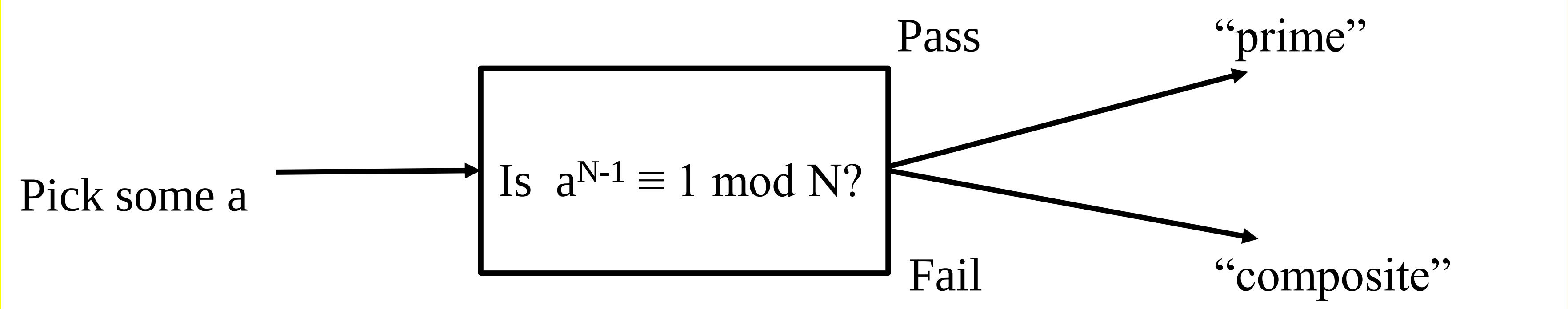
$$a^{p-1} \equiv 1 \pmod{p}.$$

i.e., $\gcd(a^{p-1}, p) = 1$; $p \mid (a^{p-1} - 1)$; $a^{p-1} \bmod p = 1 \bmod p$.

Determine whether 13 is a prime.
Assume that p =13 is a prime.
Pick a = 2, such that $1 \leq 2 < 13$.
Check whether $2^{13-1} \equiv 1 \pmod{13}$.

1. $\gcd(2^{13-1}, 13) = 1$; or
2. $13 \mid 2^{13-1} - 1$; i.e., $13 \mid 4096-1$.
3. $2^{13-1} \bmod 13 = 1 \bmod 13$

This theorem suggests a "factorless" test for determining whether a number **N** is prime:



Fermat's test

The problem is that Fermat's theorem :

- is not an if-and-only-if condition; p is prime $\rightarrow a^{p-1} \equiv 1 \pmod{p}$.
- it does not say what happens when N is not prime.
- If N is not prime, Fermat's test diagram is questionable. i.e.,
  - for any N, can we say that N is prime if $a^{N-1} \equiv 1 \pmod{N}$?
- In fact, a composite number N can possibly pass Fermat's test (that is, $a^{N-1} \equiv 1 \pmod{N}$), for certain choices of a.
  - e.g., for a non-prime N = 341 = 11 * 31, $2^{340} \equiv 1 \pmod{341}$.
- But it is true that for composite N, *most* values of *a* will fail the test.
- Show $2^{340} \bmod 341 = 2^{256+64+16+4} \bmod 341$

$= (2^{256} \bmod 341 * 2^{64} \bmod 341 * 2^{16} \bmod 341 * 2^{4} \bmod 341) \bmod 341$

$= (64 * 16 * 64 * 16) \bmod 341$

$= (1024 \bmod 341 * 1024 \bmod 341) \bmod 341$

$= (1 * 1) \bmod 341 = 1$

Figure 1.7 An algorithm for testing primality.

For this algorithm, choose *a* randomly from {1, 2, … N-1}, rather than fixing an arbitrary value of *a* in advance.

```
function primality(N)
```

Input: Positive integer N

Output: yes/no

```
    Pick a positive integer a < N at random
    if a^(N-1) ≡ 1 (mod N)
    then return yes;
    else return no;
```

Test whether $N \mid a^{N-1} - 1$?

or $\gcd(a^{N-1}, N) = 1$?

or $a^{N-1} \bmod N = 1 \bmod N$?

In analyzing the behavior of this algorithm for testing primality:

- It turns out that
  - certain extremely rare composite numbers N, called *Carmichael numbers*, pass Fermat's test for *all a* relatively prime to N. [i.e., $a^{N-1} \equiv 1 \pmod{N}$]
  - On such numbers, this algorithm will fail.

Begin ++

What is Carmichael number?

- The smallest Carmichael number is 561 = 3 * 11 * 17.
- It is not a prime.
- It passes the Fermat test, because $a^{560} \equiv 1 \pmod{561}$ for all values of a relatively prime to 561, if a $is\ not\ one\ of\ \{3, 11, 17\}$. [i.e., gcd(a, 561) = 1.]
- The numbers of this type are infinite but exceedingly rare.

There is a way around Carmichael numbers [Rabin and Miller].

- Write N – 1 in the form 2^t u.
- Choose a random base a and check the value a^{N-1} mod N.
- Perform this computation of a^{N-1} mod N by
  - first determining a^u mod N and then
  - repeatedly squaring, to get this sequence:

$$a^u \bmod N,\ a^{2u} \bmod N,\ a^{2^2 u} \bmod N,\ \ldots,\ \ a^{2^i u} = a^{N-1} \bmod N.$$

- If $a^{N-1} \not\equiv 1 \bmod N$ [i.e., the value of (a^{N-1} mod N) is not the value of 1 mod N], then N is composite by Fermat's theorem and we are done.

There is a way around Carmichael numbers [Rabin and Miller] - continued

- If $a^{N-1} \equiv 1 \pmod{N}$, we conduct a follow-up test:
  - somewhere in the preceding sequence, we ran into a 1 for the first time.
  - If this happened after the first position (that is, if $a^u \bmod N \neq 1$), and

    if the preceding value in the list is not -1 mod N,

    then we declare N composite.
    - In the latter case, a nontrivial square root of 1 modulo N is found:
      - a number that is not ±1 mod N but that when squared is equal to 1 mod N. Such a number can only exist if N is composite.
  - If we combine this square-root check with earlier Fermat test, then at least three-fourths of the possible values of a between 1 and N – 1 will reveal a composite N, even if it is a Carmichael number.

The Miller-Rabin algorithm for primality testing of integers is

- an impressive randomized algorithm (e.g., Cormen: p 968-975).
  - This randomized algorithm solves the problem
    - in an acceptable amount of time for thousand-digit numbers
    - with the probability of yielding an erroneous answer smaller than the probability of hardware malfunction.
  - It is faster than the best known deterministic algorithms for solving this problem, which is crucial for modern cryptology.

++end

In a Carmichael-free universe, the algorithm works well.

- Any prime number N will pass Fermat's test and produce the right answer.
- Any non-Carmichael composite number N must fail Fermat's test for some value of a; and
- this implies immediately that N fails Fermat's test for *at least half the possible values of a!*

Theorem 0.3:

There are infinitely many primes.

Reason: Assume that there are only n primes, $p_1, p_2, \ldots, p_n$. Let $Q = p_1 p_2 \ldots p_n + 1$. If p_k divides Q, then p_k divides $Q - p_1 p_2 \ldots p_n = 1$. This implies that Q is either a prime or a prime factor of Q.

Lemma 0.5:

If $a^{N-1} \not\equiv 1 \pmod{N}$ for some a relatively prime to N, then it must hold for at least half the choices of $a < N$.

Reason: Every $b < N$ that passes Fermat's test with respect to N (i.e., $b^{N-1} \equiv 1 \pmod{N}$) has a twin a.b, that fails the test:

$$(a.b)^{N-1} \equiv a^{N-1} . b^{N-1} \equiv a^{N-1} \not\equiv 1 \pmod{N} .$$

Disregarding the Carmichael numbers, let assert

- if N is prime, then $a^{N-1} \equiv 1 \pmod{N}$, for all $a < N$.
- if N is not prime then $a^{N-1} \equiv 1 \pmod{N}$, for at most half the values of $a < N$.

The algorithm of Figure 1.7, therefore, has the following probabilistic behavior.

- Pr(Algorithm 1.7 returns yes when N is prime) = 1
- Pr(Algorithm 1.7 returns yes when N is not prime) $\leq \frac{1}{2}$.

Reduce this one-sided error by repeating the procedure many times, by randomly picking several values of *a* and testing them all (Figure 1.8).

Pr(Algorithm 1.8 returns yes when N is not prime) $\leq \frac{1}{2^k}$.

Figure 1.8 An algorithm for testing primality, with low error probability.

```
function primality2( N )
```

Input: Positive integer N

Output: yes/no

Pick positive integers $a_1, a_2, \ldots, a_k < N$ at random

if $a_i^{N-1} \equiv 1 \pmod{N}$, for all i = 1, 2, …, k:

```
then
    return yes;
else
    return no;
```

Generating random primes

Can we have a fast algorithm for choosing random primes that are a few hundred bits long? Since primes are abundant, a random n-bit number has roughly a one-in-n chance of being prime (actually about $1/(\ln 2^n) \approx 1.44/n$). For instance, 1 in 20 social security numbers is prime! [i.e., $1/(\ln 2^{20}) = 1/20$.]

Lagrange's prime number theorem

Let $\pi(x)$ be the number of primes $\leq$ x. Then $\pi(x) \approx x / (\ln x)$, or more precisely

$$\lim_{x \to \alpha} \frac{\pi(x)}{(x/\ln x)} = 1.$$ [Note that ln x is the natural algorithm of x.]

Such abundance makes it simple to generate a random n-bit prime:

- Pick a random n-bit number N.
- Run a primality test on N.
- If it passes the test, output N; else repeat the process.

How fast is this algorithm?

If the randomly chosen N is truly prime, with a probability of at least 1/n, then it will certainly pass the test.

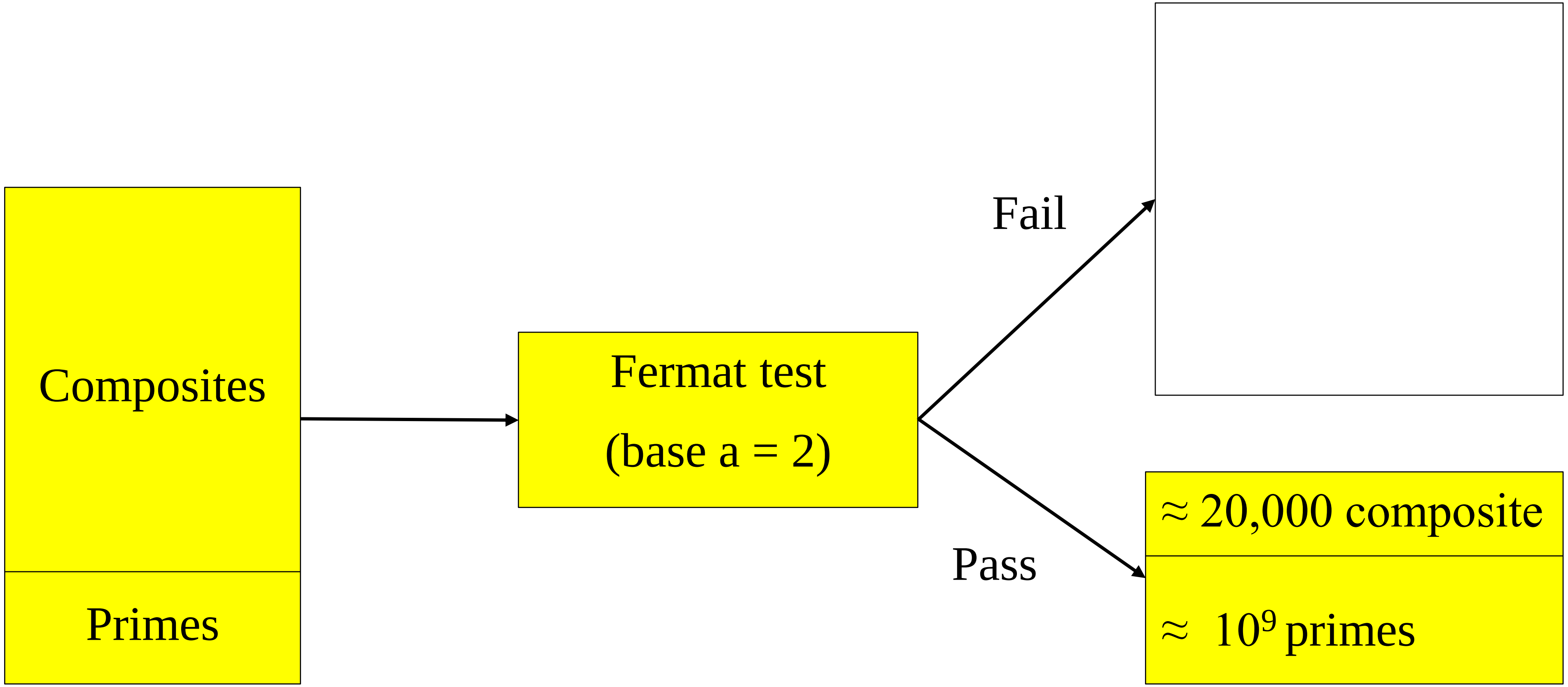
On each iteration, this procedure has at least a 1/n chance of halting. Therefore, on average it will halt within O(n) rounds.

Which primality test should be used?

It is sufficient to perform the Fermat test with base a = 2 (or to be really safe, a = 2, 3, 5) because for random numbers the Fermat test has a much smaller failure probability than the worst-case ½ bound.

What is the probability that the output of the algorithm is really prime?

- Suppose the test is performed with base a = 2 for all numbers N $\leq$ 25 * 10^9.
- In this range, there are about 10^9 primes, and about 20,000 composites that pass the test.
- Thus, the chance of erroneously outputting a composite is approximately 20,000/10^9 = 2 * 10^{-5}.
- This chance of error decreases rapidly as the length of the numbers involved is increased to a few hundred digits we expect in applications.



Before primality test:

All numbers $N \leq 25 * 10^9$

after primality test

Finding Large Prime Numbers

- find large prime numbers, which is necessary to the success of the RSA public-key cryptosystem.
- then show an algorithm for testing whether a number is prime.

Search of a Large Prime

To find a large prime number,

- select randomly integers of the appropriate size and test whether each selected integer is prime until one is found to be prime.
- An important consideration of this method is
  - the likelihood of finding a prime when an integer is chosen at random.

Give the prime distribution theorem, which enables us to approximate this likelihood.

The prime distribution function $\pi(n)$ is

- the number of primes that are less than or equal to n.
- For example, $\pi(12) = 5$ since there are five primes, 2, 3, 5, 7 and 11, that are less than or equal to 12.
- The prime number theorem show in Theorem 0.15 gives an approximation of $\pi(n)$.

Search of a Large Prime

Theorem 0.15 The prime distribution theorem - *Lagrange's prime number theorem*

We have that

$$\lim_{n \to \infty} \frac{\pi(n)}{\frac{n}{\ln n}} = 1.$$

If we randomly choose an integer between 1 and $n = 10^{16}$ according to the uniform distribution, the probability of it being prime is about

$$\frac{1}{\ln 10^{16}} = 0.027143.$$

Suppose we choose 200 such numbers at random. The probability of them all not being prime is then about

$$(1 - 0.027143)^{200} = 0.004.$$

If we randomly choose an integer between 1 and $n = 10^{100}$ according to the uniform distribution, the probability of it being prime is about

$$\frac{1}{\ln 10^{100}} = 0.0043429.$$

Suppose we choose 200 such numbers at random. The probability of them all not being prime is then about

$$(1 - 0.0043429)^{200} = 0.04.$$

Cryptography – The RSA Public Key Cryptosystem

The Rivest-Shamir-Adleman (RSA) cryptosystem uses all the ideas we have introduced in this lecture note. It derives very strong guarantees of security by ingeniously exploiting the wide gulf between the polynomial-time computability of certain number-theoretic tasks: (

- modular exponentiation,
- greatest common divisor,
- primality testing) and
- the intractability of others (factoring).