DISCRETE MATHEMATICS

INTEGERS AND DIVISION

Integers and division

- **Number theory** is a branch of mathematics that explores integers and their properties.
- **Integers:**
  - **Z integers {…, -2,-1, 0, 1, 2, …}**
  - **Z^+ positive integers {1, 2, …}**

- Number theory has many applications within computer science, including:
  - Indexing - Storage and organization of data
  - Encryption
  - Error correcting codes
  - Random numbers generators

Division

Definition: Assume 2 integers a and b, such that a =/ 0 (a is not equal 0). We say that **a divides b** if there is an integer c such that b = ac. If a divides b we say that **a is a *factor* of b** and that **b is *multiple* of a**.

- The fact that a divides b is denoted as **a | b.**

Examples:

- 4 | 24 True or False ? **True**
  - 4 is a factor of 24
  - 24 is a multiple of 4
- 3 | 7 True or False ? **False**

Divisibility

Properties:

- Let a, b, c be integers. Then the following hold:
  1. if a | b and a | c then a | (b +c)
  2. if a | b then a | bc for all integers c
  3. if a | b and b | c then a | c

Proof of 1: if a | b and a | c then a | (b +c)

- from the definition of divisibility we get:
- b=au and c=av where u,v are two integers. Then
- (b+c) = au +av = a(u+v)
- **Thus a divides b+c.**

Divisibility

Properties:

- Let a, b, c be integers. Then the following hold:
  1. if a | b and a | c then a | (b +c)
  2. if a | b then a | bc for all integers c
  3. if a | b and b | c then a | c

Proof of 2: if a | b then a | bc for all integers c

- If a | b, then there is some integer u such that b = au.
- Multiplying both sides by c gives us bc = auc, so by definition, a | bc.
- **Thus a divides bc.**

Primes

Definition: A positive integer p that is greater than 1 and that is divisible only by 1 and by itself (p) is called **a prime**.

Examples: 2, 3, 5, 7, …

1 | 2 and 2 | 2, 1 |3 and 3 | 3, etc

Primes

Definition: A positive integer p that is greater than 1 and that is divisible only by 1 and by itself (p) is called **a prime**.

Examples: 2, 3, 5, 7, …

1 | 2 and 2 | 2, 1 |3 and 3 | 3, etc

What is the next prime after 7?

- 11

Next?

- 13

Primes

Definition: A positive integer that is greater than 1 and is not a prime is called **a composite.**

Examples: 4, 6, 8, 9, …

Why?

2 | 4

Why 6 is a composite?

The Fundamental theorem of Arithmetic

Fundamental theorem of Arithmetic:

- Any positive integer greater than 1 can be expressed as a product of prime numbers.

Examples:

- $12 = ?$

The Fundamental theorem of Arithmetic

Fundamental theorem of Arithmetic:

- Any positive integer greater than 1 can be expressed as a product of prime numbers.

Examples:

- $12 = 2*2*3$
- $21 = 3*7$

- Process of finding out factors of the product: **factorization**.

Primes and composites

Factorization of composites to primes:

- $100 = 2*2*5*5 = 2^2*5^2$
- $99 = 3*3*11 = 3^2*11$

Important question:

- How to determine whether the number is a prime or a composite?

Primes and composites

- How to determine whether the number is a prime or a composite?

Simple approach (1):

- Let n be a number. To determine whether it is a prime we can test if any number $x < n$ divides it. If yes it is a composite. If we test all numbers $x < n$ and do not find the proper divisor then n is a prime.

Primes and composites

- How to determine whether the number is a prime or a composite?

Simple approach (1):

- Let n be a number. To determine whether it is a prime we can test if any number $x < n$ divides it. If yes it is a composite. If we test all numbers $x < n$ and do not find the proper divisor then n is a prime.

- **Example:**
- Assume we want to check if 17 is a prime?
- The approach would require us to check:
- 2,3,4,5,6,7,8,9,10,11,12,13,14,15,16

Primes and composites

- **Example approach 1:**
- Assume we want to check if 17 is a prime?
- The approach would require us to check:
- 2,3,4,5,6,7,8,9,10,11,12,13,14,15,16

- **Is this the best we can do?**
- **No.** The problem here is that we try to test all the numbers. But this is not necessary.
- **Idea:** Every composite factorizes to a product of primes. So it is sufficient to test only the primes $x < n$ to determine the primality of n.

Primes and composites

- How to determine whether the number is a prime or a composite?

Approach 2:

- Let *n* be a number. To determine whether it is a prime we can test if any prime number x < n divides it. If yes it is a composite. If we test all primes $x < n$ and do not find a proper divisor then *n* is a prime.

Primes and composites

- How to determine whether the number is a prime or a composite?

Approach 2:

- Let *n* be a number. To determine whether it is a prime we can test if any prime number x < n divides it. If yes it is a composite. If we test all primes $x < n$ and do not find a proper divisor then *n* is a prime.

- **Example:** Is 31 a prime?
- Check if 2,3,5,7,11,13,17,23,29 divide it
- It is a prime !!

Primes and composites

Example approach 2:

Is 91 a prime number?

- Easy primes 2,3,5,7,11,13,17,19 ..
- But how many primes are there that are smaller than 91?

Caveat:

- If *n* is relatively small the test is good because we can enumerate (memorize) all small primes
- But if *n* is large there can be larger not obvious primes

Primes and composites

Theorem: If n is a composite then n has a prime divisor less than or equal to $\sqrt{n}$.

Primes and composites

Theorem: If n is a composite then n has a prime divisor less than or equal to $\sqrt{n}$.

Proof:

- If n is composite, then it has a positive integer factor a such that $1 < a < n$ by definition. This means that $n = ab$, where b is an integer greater than 1.
- Assume $a > \sqrt{n}$ and $b > \sqrt{n}$. Then $ab > \sqrt{n}\sqrt{n} = n$, which is a contradiction. So either $a \leq \sqrt{n}$ *or* $b \leq \sqrt{n}$.
- Thus, n has a divisor less than $\sqrt{n}$.
- By the fundamental theorem of arithmetic, this divisor is either prime, or is a product of primes. In either case, n has a prime divisor less than $\sqrt{n}$.

Primes and composites

Theorem: If n is a composite that n has a prime divisor less than or equal to $\sqrt{n}$.

Approach 3:

- Let *n* be a number. To determine whether it is a prime we can test if any prime number x < $\sqrt{n}$ divides it.

Example 1: Is 101 a prime?

- Primes smaller than $\sqrt{101} = 10.xxx$ are: 2,3,5,7
- 101 is not divisible by any of them
- **Thus 101 is a prime**

Example 2: Is 91 a prime?

- Primes smaller than $\sqrt{91}$ are: 2,3,5,7
- 91 is divisible by 7
- **Thus 91 is a composite**

Primes

Question: How many primes are there?

Theorem: There are infinitely many primes.

Primes

Question: How many primes are there?

Theorem: There are infinitely many primes.

Proof by Euclid.

- Proof by contradiction:
  - Assume there is a finite number of primes: $p_1, p_2, \ldots p_n$
- Let $Q = p_1 p_2 \ldots p_n + 1$ be a number.
- None of the numbers $p_1, p_2, \ldots, p_n$ divides the number Q.
- This is a contradiction since we assumed that we have listed all primes.

Division

Let a be an integer and d a positive integer. Then there are unique integers, q and r, with $0 <= r < d$, such that

$$a = dq + r.$$

Definitions:

- a is called the **dividend**,
- d is called the **divisor**,
- q is called the **quotient** and
- r the **remainder** of the division.

Example: a= 14, d = 3

14 = 3*4 + 2

14/3=3.666

14 div 3 = 4

14 mod 3 = 2

Relations:

- **q = a div d , r = a mod d**

Greatest common divisor

Definition: Let a and b are integers, not both 0. Then the largest integer d such that d | a and d | b is called **the greatest common divisor** of a and b. The greatest common divisor is denoted as gcd(a,b).

Examples:

- gcd(24,36) = ?
- Check 2,3,4,6,12 gcd(24,36) = 12
- gcd(11,23) = ?

Greatest common divisor

A systematic way to find the gcd using factorization:

- Let $a=p_1^{a1} p_2^{a2} p_3^{a3} \dots p_k^{ak}$ and $b= p_1^{b1} p_2^{b2} p_3^{b3} \dots p_k^{bk}$
- $\gcd(a,b)= p_1^{\min(a1,b1)} p_2^{\min(a2,b2)} p_3^{\min(a3,b3)} \dots p_k^{\min(ak,bk)}$

Examples:

- gcd(24,36) = ?
- $24 = 2*2*2*3=2^3*3$
- $36= 2*2*3*3=2^2*3^2$
- $\gcd(24,36) =2^2*3 =$ **12**

Least common multiple

Definition: Let a and b are two positive integers. The least common multiple of a and b is the smallest positive integer that is divisible by both a and b. The **least common multiple** is denoted as **lcm(a,b).**

Example:

- **What is lcm(12,9) =?**
- Give me a common multiple: …

Least common multiple

Definition: Let a and b are two positive integers. The least common multiple of a and b is the smallest positive integer that is divisible by both a and b. The **least common multiple** is denoted as **lcm(a,b).**

Example:

- **What is lcm(12,9) =?**
- Give me a common multiple: … 12*9= 108
- **Can we find a smaller number?**

Least common multiple

Definition: Let a and b are two positive integers. The least common multiple of a and b is the smallest positive integer that is divisible by both a and b. The **least common multiple** is denoted as **lcm(a,b).**

Example:

- **What is lcm(12,9) =?**
- Give me a common multiple: … 12*9= 108
- **Can we find a smaller number?**
- **Yes.** Try 36. Both 12 and 9 cleanly divide 36.

Least common multiple

A systematic way to find the lcm using factorization:

- Let $a=p_1^{a1} p_2^{a2} p_3^{a3} \dots p_k^{ak}$ and $b= p_1^{b1} p_2^{b2} p_3^{b3} \dots p_k^{bk}$
- $lcm(a,b)= p_1^{\max(a1,b1)} p_2^{\max(a2,b2)} p_3^{\max(a3,b3)} \dots p_k^{\max(ak,bk)}$

Example:

- **What is lcm(12,9) =?**
- $12 = 2*2*3=2^2*3$
- $9=3*3 =3^2$
- **lcm(12,9)** $= 2^2 * 3^2 = 4 * 9 =$ **36**

Euclid algorithm

Finding the greatest common divisor requires factorization

- $a=p_1^{a1} p_2^{a2} p_3^{a3} \dots p_k^{ak}$, $b= p_1^{b1} p_2^{b2} p_3^{b3} \dots p_k^{bk}$
- $\gcd(a,b)= p_1^{\min(a1,b1)} p_2^{\min(a2,b2)} p_3^{\min(a3,b3)} \dots p_k^{\min(ak,bk)}$

- Factorization can be cumbersome and time consuming since we need to find all factors of the two integers that can be very large.

- Luckily a more efficient method for computing the gcd exists:
- It is called **Euclid's algorithm**
  - the method is known from ancient times and named after Greek mathematician Euclid.

Euclid algorithm

Assume two numbers 287 and 91. We want gcd(287,91).

- First divide the larger number (287) by the smaller one (91)
- We get 287 = 3*91 +14

(1) Any divisor of 91 and 287 must also be a divisor of 14:

  - 287 - 3*91 = 14
- Why? [ak – cbk] =r → (a-cb)k = r → (a-cb) = r/k (must be an integer and thus k divides r]

(2) Any divisor of 91 and 14 must also be a divisor of 287

- Why? 287 = 3 b k + dk → 287 = k(3b +d) → 287 /k = (3b +d) ← 287/k must be an integer
- **But then gcd(287,91) = gcd(91,14)**

Euclid algorithm

- **We know that gcd(287,91) = gcd(91,14)**
- But the same trick can be applied again:
    - gcd(91,14)
    - 91 = 14.6 + 7
- and therefore
    - gcd(91,14)=gcd(14,7)

- And one more time:
    - gcd(14,7) = 7
    - trivial
- **The result: gcd(287,91) = gcd(91,14)=gcd(14,7) = 7**

Euclid algorithm

Example 1:

- Find the greatest common divisor of 666 & 558

- gcd(666,558) 666=1*558 + 108
  = gcd(558,108) 558=5*108 + 18
  = gcd(108,18) 108=6*18 + 0
  = **18**

Euclid algorithm

Example 2:

- Find the greatest common divisor of 286 & 503:

- gcd(503,286) 503=

Euclid algorithm

Example 2:

- Find the greatest common divisor of 286 & 503:

- gcd(503,286) 503=1*286 + 217
  =gcd(286, 217) 286=

Euclid algorithm

Example 2:

- Find the greatest common divisor of 286 & 503:

- gcd(503,286)
  =gcd(286, 217)
  =gcd(217, 69)
  = gcd(69,10)
  =gcd(10,9)
  = gcd(9,1) =**1**

503=1*286 + 217
286=1*217 + 69
217 = 3*69 + 10
69 = 6*10 + 9
10=1*9 + 1

Modular arithmetic

- In computer science we often care about the remainder of an integer when it is divided by some positive integer.

Problem: Assume that it is a midnight. What is the time on the 24 hour clock after 50 hours?

Answer: the result is 2am

How did we arrive to the result:

- Divide 50 with 24. The reminder is the time on the 24 hour clock.
  - 50= 2*24 + 2
  - so the result is 2am.

Congruency

Definition: If a and b are integers and m is a positive integer, then **a is congruent to b modulo n** if m divides a-b. We use the notation **a = b (mod m)** to denote the congruency. If a and b are not congruent we write $a \neq b \pmod{m}$.

Example:

- Determine if 17 is congruent to 5 modulo 6?

Congruency

Theorem. If a and b are integers and m a positive integer. Then a=b (mod m) if and only if a mod m = b mod b.

Example:

- Determine if 17 is congruent to 5 modulo 6?
- 17 mod 6 = **5**
- 5 mod 6 = **5**
- Thus 17 is congruent to 5 modulo 6.

Congruencies

Theorem 1. Let m be a positive integer. The integers a and b are congruent modulo m if and only if there exists an integer k such that a=b+mk.

Theorem2 . Let m be a positive integer. If a=b (mod m) and c=d (mod m) then:

a+c = b+d (mod m) and ac=bd (mod m).

COROLLARY

Let m be a positive integer and let a and b be integers. Then

$$(a + b) \bmod m = ((a \bmod m) + (b \bmod m)) \bmod m$$

and

$$ab \bmod m = ((a \bmod m)(b \bmod m)) \bmod m.$$

Proof: By the definitions of $\bmod m$ and of congruence modulo m, we know that $a \equiv (a \bmod m) \pmod m$ and $b \equiv (b \bmod m) \pmod m$. Hence, Theorem 5 tells us that

$$a + b \equiv (a \bmod m) + (b \bmod m) \pmod m$$

and

$$ab \equiv (a \bmod m)(b \bmod m) \pmod m.$$

The equalities in this corollary follow from these last two congruences by Theorem 3. ◁

Modular arithmetic in CS

Modular arithmetic and congruencies are used in CS:

- **Pseudorandom number generators**
- **Hash functions**
- **Cryptology**

Pseudorandom number generators

- **Some problems we want to program need to simulate a random choice.**
- **Examples: flip of a coin, roll of a dice**

We need a way to generate random outcomes

Basic problem:

- assume outcomes: 0, 1, .. N
- generate the random sequences of outcomes

- Pseudorandom number generators let us generate sequences that look random
- **Next:** linear congruential method

Pseudorandom number generators

Linear congruential method

- We choose 4 numbers:
    - the modulus m,
    - multiplier a,
    - increment c, and
    - seed x_0,

  such that $2 =< a < m$, $0 =< c < m$, $0 =< x_0 < m$.

- We generate a sequence of numbers $x_1, x_2 x_3 \ldots x_n \ldots$ such that $0 =< x_n < m$ for all *n* by successively using the congruence:
    - $x_{n+1} = (a.x_n + c) \bmod m$

Pseudorandom number generators

Linear congruential method:

- $x_{n+1} = (a.x_n + c) \bmod m$

Example:

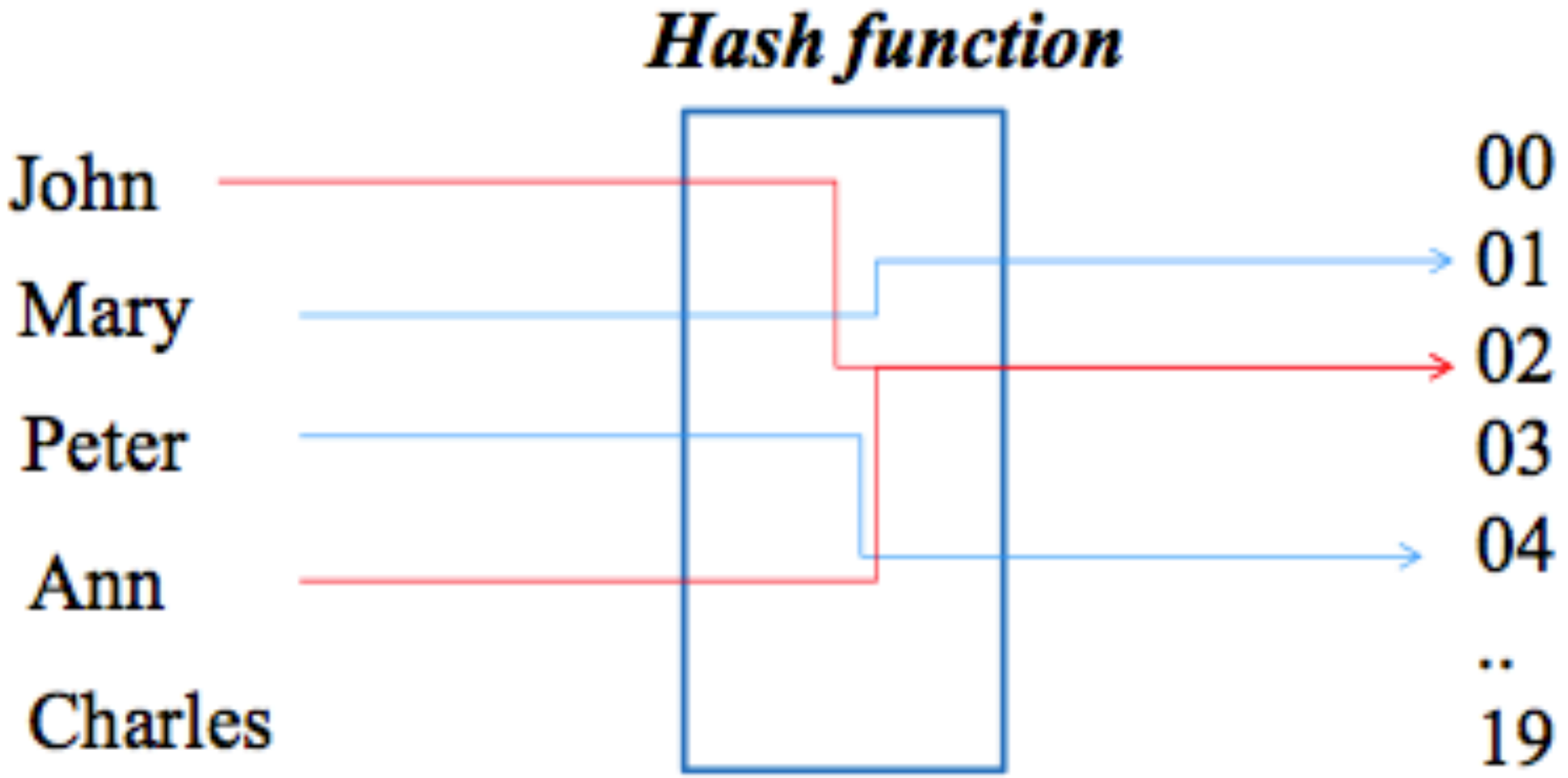
- Assume : m=9,a=7,c=4, $x_0 = 3$
- $x_1 = 7*3+4 \bmod 9 = 25 \bmod 9 = 7$
- $x_2 = 53 \bmod 9 = 8$
- $x_3 = 60 \bmod 9 = 6$
- $x_4 = 46 \bmod 9 = 1$
- $x_5 = 11 \bmod 9 = 2$
- $x_6 = 18 \bmod 9 = 0$
- ....

Hash functions

A ***hash function*** is an algorithm that maps data of arbitrary length to data of a fixed length.

The values returned by a hash function are called **hash values** or **hash codes.**

Example:



Hash function

An example of a hash function that maps integers (including very large ones) to a subset of integers 0, 1, .. m-1 is:

h(k) = k mod m

Example: Assume we have a database of employes, each with a unique ID – a social security number that consists of 8 digits. We want to store the records in a smaller table with m entries. Using h(k) function we can map a social secutity number in the database of employes to indexes in the table.

Assume: h(k) = k mod 111

Then:

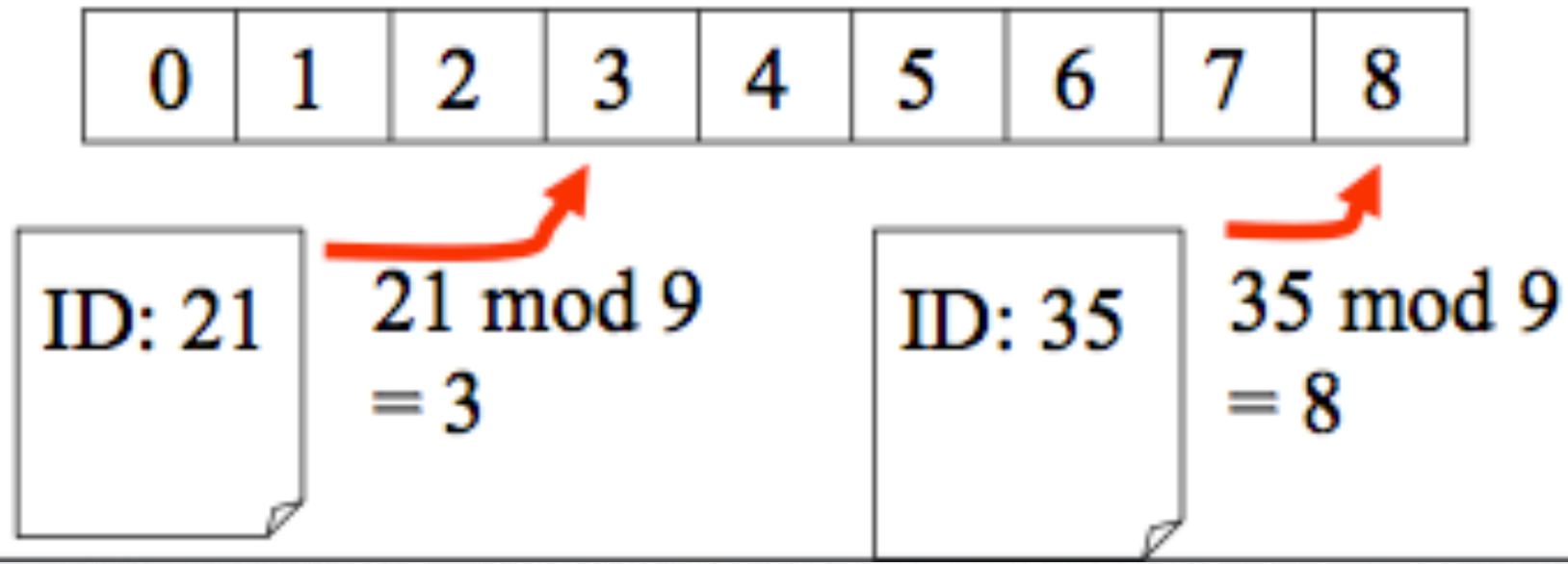
h(064212848) = 064212848 mod 111 = 14

h(037149212) = 037149212 mod 111 = 65

Hash functions

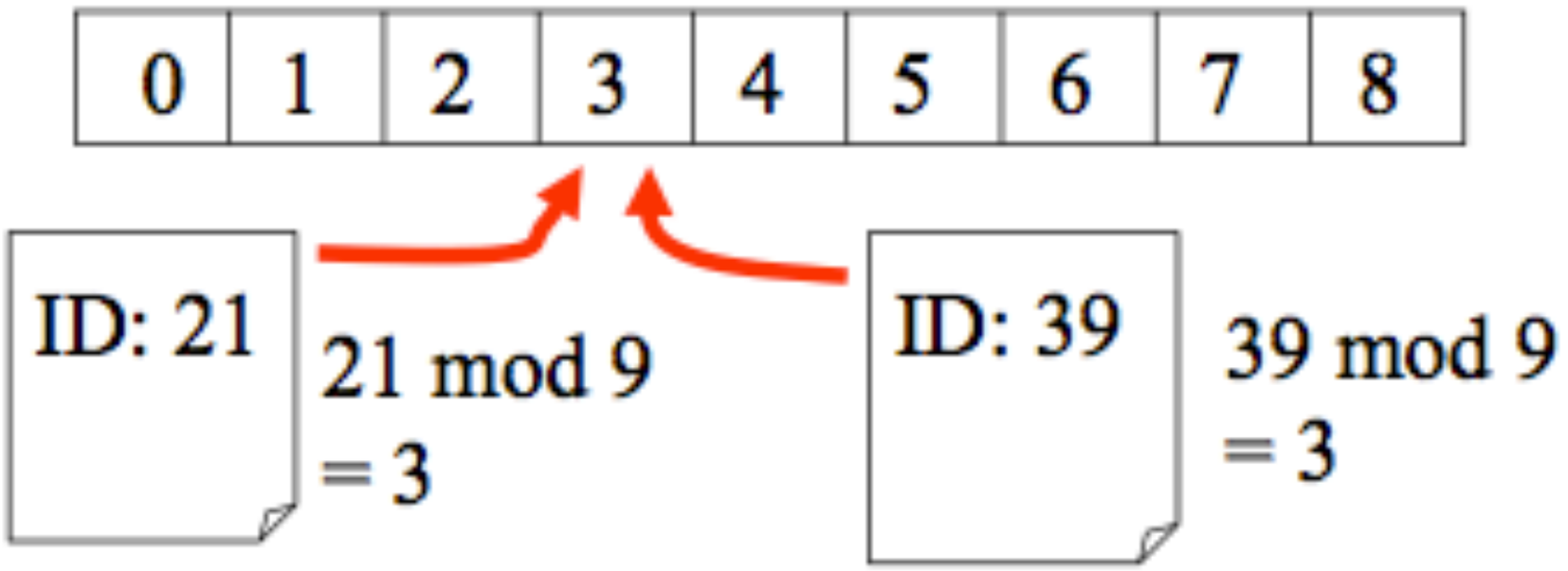
- **Problem:** Given a large collection of records, how can we store and find a record quickly?
- **Solution:** Use a hash function calculate the location of the record based on the record's ID.
- **Example:** A common hash function is
  - $h(k) = k \bmod n$,

where n is the number of available storage locations.



Hash functions

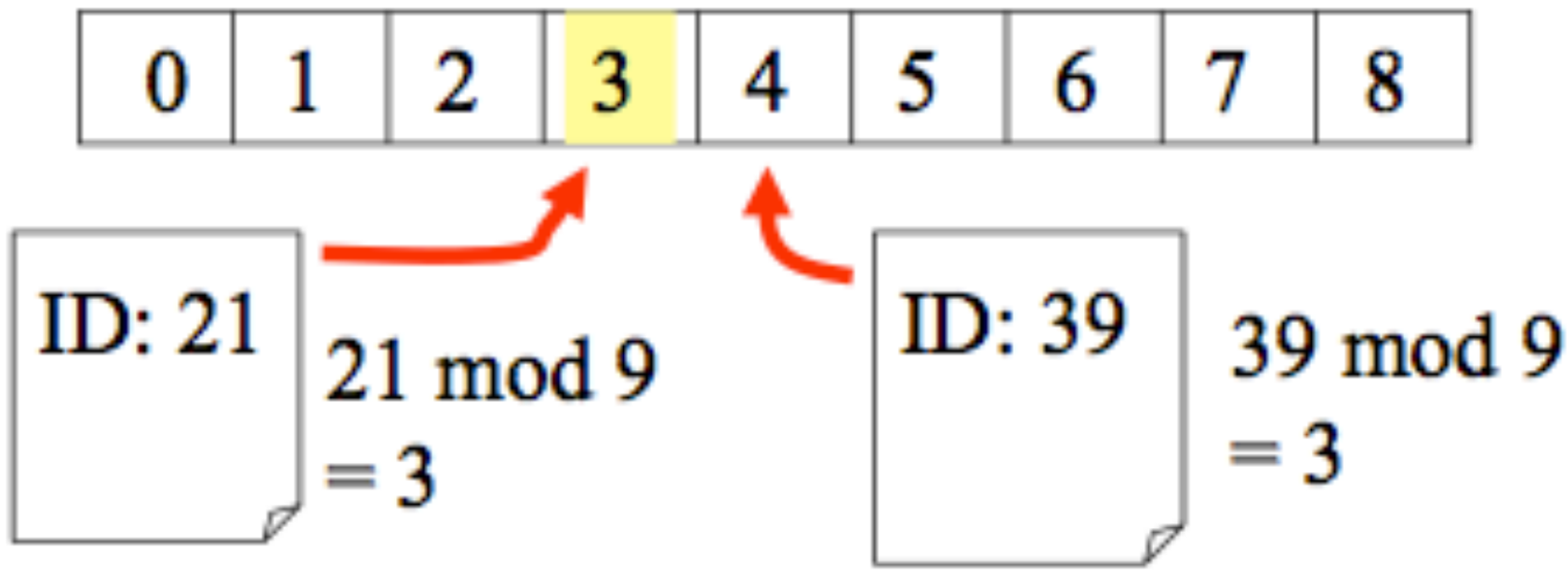
- **Problem:** two documents mapped to the same location



Hash functions

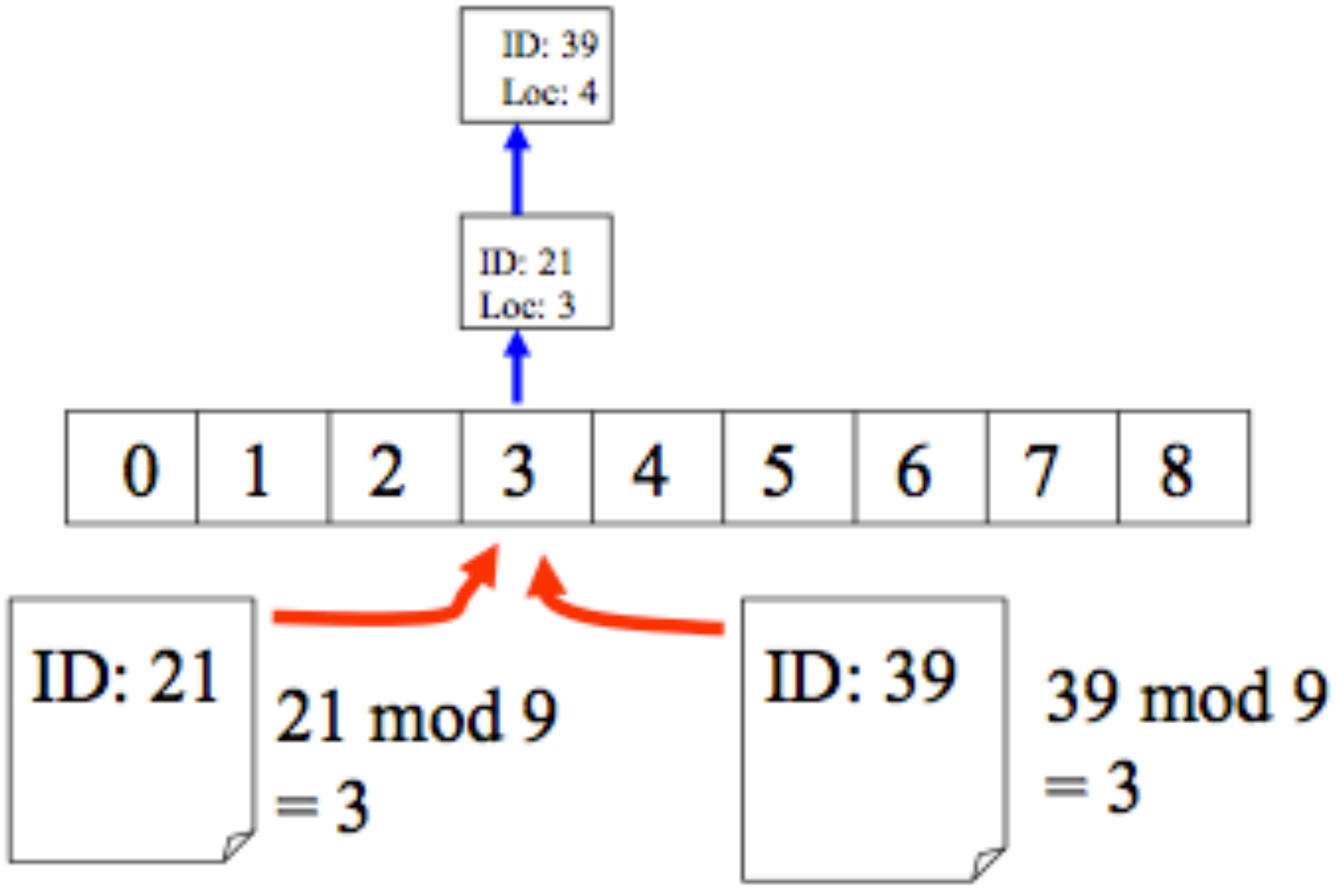
- **Solution 1:** move the next available location
  - Method is represented by a sequence of hash functions to try

$$h_0(k) = k \bmod n$$
$$h_1(k) = (k+1) \bmod n$$
...
$$h_m(k) = (k+m) \bmod n$$



Hash functions

- **Solution 2:** remember the exact location in a secondary structure that is searched sequentially



Cryptology

Encryption of messages.

- **Ceasar cipher:**
- Shift letters in the message by 3, last three letters mapped to the first 3 letters, e.g. A is shifted to D, X is shifted to A

How to represent the idea of a shift by 3?

- There are 26 letters in the alphabet. Assign each of them a number from 0,1, 2, 3, .. 25 according to the alphabetical order.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	Y	V	X	W	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

- The encryption of the letter with an index p is represented as:
  - $f(p) = (p + 3) \bmod 26$

Cryptology

Encryption of messages using a shift by 3.

- The encryption of the letter with an index p is represented as:
  - $f(p) = (p + 3) \bmod 26$

Coding of letters:

A B C D E F G H I J K L M N O P Q R S T U Y V X W Z

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

- **Encrypt message:**
  - **I LIKE DISCRETE MATH**
  -

Cryptology

Encryption of messages using a shift by 3.

- The encryption of the letter with an index p is represented as:
  - $f(p) = (p + 3) \bmod 26$

Coding of letters:

A B C D E F G H I J K L M N O P Q R S T U Y V X W Z

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

- **Encrypt message:**
  - **I LIKE DISCRETE MATH**
  - **L 0LNH GLYFUHVH PDVK.**

Cryptology

How to decode the message ?

- The encryption of the letter with an index p is represented as:
  - $f(p) = (p + 3) \bmod 26$

Coding of letters:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	Y	V	X	W	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

- **What method would you use to decode the message:**
  - **$f^{-1}(p) = (p-3) \bmod 26$**

CRYPTOSYSTEMS

A *cryptosystem* is a five-tuple $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, where $\mathcal{P}$ is the set of plaintext strings, $\mathcal{C}$ is the set of ciphertext strings, $\mathcal{K}$ is the *keyspace* (the set of all possible keys), $\mathcal{E}$ is the set of encryption functions, and $\mathcal{D}$ is the set of decryption functions. We denote by E_k the encryption function in $\mathcal{E}$ corresponding to the key k and D_k the decryption function in $\mathcal{D}$ that decrypts ciphertext that was encrypted using E_k, that is $D_k(E_k(p)) = p$, for all plaintext strings p.

PUBLIC KEY CRYPTOGRAPHY

- Shift cipher is an example of **private key cryptosystems**
  - With an encryption key, you can quickly find the decryption key.
    - Need to securely exchange this key.
- Public key cryptosystems as an alternative
  - Introduced in 1970s
  - A publicly known encryption key vs. secret decryption keys
    - An extraordinary amount of work (such as billions of years of computer time) needed to recover the plaintext message without knowing decryption keys.

THE RSA CRYPTOSYSTEM

- Ronald Rivest, Adi Shamir and Leonard Adleman (1976) from the Massachusetts Institute of Technology.

- Each individual has an encryption key (n, e)

  - n (the modulus): the product of two large primes p and q, say with 200 digits each.
  - e: the relatively prime to $(p - 1)(q - 1)$

MODULAR ARITHMETIC IN RSA

- Great common divisors as Linear Combinations

BÉZOUT'S THEOREM If a and b are positive integers, then there exist integers s and t such that $\gcd(a, b) = sa + tb$.

EXAMPLE 17 Express $\gcd(252, 198) = 18$ as a linear combination of 252 and 198.

Solution: To show that $\gcd(252, 198) = 18$, the Euclidean algorithm uses these divisions:

$$252 = 1 \cdot 198 + 54$$
$$198 = 3 \cdot 54 + 36$$
$$54 = 1 \cdot 36 + 18$$
$$36 = 2 \cdot 18.$$

Using the next-to-last division (the third division), we can express $\gcd(252, 198) = 18$ as a linear combination of 54 and 36. We find that

$$18 = 54 - 1 \cdot 36.$$

The second division tells us that

$$36 = 198 - 3 \cdot 54.$$

Substituting this expression for 36 into the previous equation, we can express 18 as a linear combination of 54 and 198. We have

$$18 = 54 - 1 \cdot 36 = 54 - 1 \cdot (198 - 3 \cdot 54) = 4 \cdot 54 - 1 \cdot 198.$$

GREAT COMMON DIVISORS AS LINEAR COMBINATIONS

If a, b, and c are positive integers such that $\gcd(a, b) = 1$ and $a \mid bc$, then $a \mid c$.

Proof: Because $\gcd(a, b) = 1$, by Bézout's theorem there are integers s and t such that

$$sa + tb = 1.$$

Multiplying both sides of this equation by c, we obtain

$$sac + tbc = c.$$

We can now use Theorem 1 of Section 4.1 to show that $a \mid c$. By part (ii) of that theorem, $a \mid tbc$. Because $a \mid sac$ and $a \mid tbc$, by part (i) of that theorem, we conclude that a divides $sac + tbc$. Because $sac + tbc = c$, we conclude that $a \mid c$, completing the proof. ◁

GREAT COMMON DIVISORS AS LINEAR COMBINATIONS

Let m be a positive integer and let a, b, and c be integers. If $ac \equiv bc \pmod{m}$ and $\gcd(c, m) = 1$, then $a \equiv b \pmod{m}$.

Proof: Because $ac \equiv bc \pmod{m}$, $m \mid ac - bc = c(a - b)$. By Lemma 2, because $\gcd(c, m) = 1$, it follows that $m \mid a - b$. We conclude that $a \equiv b \pmod{m}$. ◁

SOLVING LINEAR CONGRUENCES

If a and m are relatively prime integers and $m > 1$, then an inverse of a modulo m exists. Furthermore, this inverse is unique modulo m. (That is, there is a unique positive integer $\overline{a}$ less than m that is an inverse of a modulo m and every other inverse of a modulo m is congruent to $\overline{a}$ modulo m.)

Proof: By Theorem 6 of Section 4.3, because $\gcd(a, m) = 1$, there are integers s and t such that

$$sa + tm = 1.$$

This implies that

$$sa + tm \equiv 1 \pmod{m}.$$

Because $tm \equiv 0 \pmod{m}$, it follows that

$$sa \equiv 1 \pmod{m}.$$

Consequently, s is an inverse of a modulo m. That this inverse is unique modulo m is left as Exercise 7. ◁

PRIMITIVE ROOTS AND DISCRETE LOGARITHMS

A *primitive root* modulo a prime p is an integer r in $\mathbf{Z}_p$ such that every nonzero element of $\mathbf{Z}_p$ is a power of r.

Determine whether 2 and 3 are primitive roots modulo 11.

Solution: When we compute the powers of 2 in $\mathbf{Z}_{11}$, we obtain $2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 5, 2^5 = 10, 2^6 = 9, 2^7 = 7, 2^8 = 3, 2^9 = 6, 2^{10} = 1$. Because every element of $\mathbf{Z}_{11}$ is a power of 2, 2 is a primitive root of 11.

When we compute the powers of 3 modulo 11, we obtain $3^1 = 3, 3^2 = 9, 3^3 = 5, 3^4 = 4, 3^5 = 1$. We note that this pattern repeats when we compute higher powers of 3. Because not all elements of $\mathbf{Z}_{11}$ are powers of 3, we conclude that 3 is not a primitive root of 11. ◀

PRIMITIVE ROOTS AND DISCRETE LOGARITHMS

Suppose that p is a prime, r is a primitive root modulo p, and a is an integer between 1 and $p-1$ inclusive. If $r^e \bmod p = a$ and $0 \leq e \leq p-1$, we say that e is the *discrete logarithm* of a modulo p to the base r and we write $\log_r a = e$ (where the prime p is understood).

Find the discrete logarithms of 3 and 5 modulo 11 to the base 2.

Solution: When we computed the powers of 2 modulo 11 in Example 12, we found that $2^8 = 3$ and $2^4 = 5$ in $\mathbf{Z}_{11}$. Hence, the discrete logarithms of 3 and 5 modulo 11 to the base 2 are 8 and 4, respectively. (These are the powers of 2 that equal 3 and 5, respectively, in $\mathbf{Z}_{11}$.) We write $\log_2 3 = 8$ and $\log_2 5 = 4$ (where the modulus 11 is understood and not explicitly noted in the notation).

IT'S TIME TO SEE **RSA**

RSA ENCRYPTION

- A plaintext message M is first translated into sequences of two-digit numbers
  - A is translated into 00, B into 01, . . . , and J into 09, ...
  - Concatenate these two-digit numbers into strings of digits
  - Divide this string into equally sized blocks of 2N digits
    - 2N: the largest even number such that the number 2525 . . . 25 with 2N digits does not exceed n
  - As a result, M is now translated into a sequence of integers *m1*, *m2*, . . . , *mk* for some integer k.
  - Transform each block *mi* to a ciphertext block *ci*
    - $C = M^e \bmod n$ *(seeAlgorithm 5 in Section 4.2. for fast modular exponentiation)*

RSA ENCRYPTION: EXAMPLE

Encrypt the message STOP using the RSA cryptosystem with key (2537, 13). Note that $2537 = 43 \cdot 59$, $p = 43$ and $q = 59$ are primes, and

$$\gcd(e, (p-1)(q-1)) = \gcd(13, 42 \cdot 58) = 1.$$

Solution: To encrypt, we first translate the letters in STOP into their numerical equivalents. We then group these numbers into blocks of four digits (because $2525 < 2537 < 252525$), to obtain

$$1819 \quad 1415.$$

We encrypt each block using the mapping

$$C = M^{13} \bmod 2537.$$

Computations using fast modular multiplication show that $1819^{13} \bmod 2537 = 2081$ and $1415^{13} \bmod 2537 = 2182$. The encrypted message is 2081 2182. ◀

RSA DECRYPTION

- The plaintext message can be quickly recovered when the decryption key d, an inverse of e modulo $(p - 1)(q - 1)$, is known, namely $de \equiv 1 \pmod{(p - 1)(q - 1)}$
  - $de = 1 + k(p - 1)(q - 1)$
  - $C^d \equiv (M^e)^d = M^{de} = M^{1+k(p-1)(q-1)} \pmod{n}$
  - It is proven that: $C^d \equiv M \pmod{pq}$

RSA DECRYPTION: EXAMPLE

We receive the encrypted message 0981 0461. What is the decrypted message if it was encrypted using the RSA cipher from Example 8?

Solution: The message was encrypted using the RSA cryptosystem with $n = 43 \cdot 59$ and exponent 13. As Exercise 2 in Section 4.4 shows, $d = 937$ is an inverse of 13 modulo $42 \cdot 58 = 2436$. We use 937 as our decryption exponent. Consequently, to decrypt a block C, we compute

$$M = C^{937} \bmod 2537.$$

To decrypt the message, we use the fast modular exponentiation algorithm to compute $0981^{937} \bmod 2537 = 0704$ and $0461^{937} \bmod 2537 = 1115$. Consequently, the numerical version of the original message is 0704 1115. Translating this back to English letters, we see that the message is HELP.

RSA AS A PUBLIC KEY SYSTEM

- RSA cryptosystem is suitable for public key cryptography
  - Possible to rapidly construct a public key by finding two large primes p and q, each with more than 200 digits
  - Possible to find an integer e relatively prime to $(p - 1)(q - 1)$
  - Quickly find an inverse d of e modulo $(p - 1)(q - 1)$
  - However, no method is known to decrypt messages that is not based on finding a factorization of n
    - The most efficient factorization methods known (as of 2010) require billions of years to factor 400-digit integers.
    - No polynomial-time algorithm is known for factoring large integers by far

CRYPTOGRAPHIC PROTOCOLS

❖ KEY EXCHANGE

- Two parties can use to exchange a secret key over an insecure communications channel without having shared any information in the past

- Diffie-Hellman key agreement protocol (1976)

Suppose that Alice and Bob want to share a common key. The protocol follows these steps, where the computations are done in $\mathbf{Z}_p$.

(1) Alice and Bob agree to use a prime p and a primitive root a of p.

(2) Alice chooses a secret integer k_1 and sends a^{k_1} **mod** p to Bob.

(3) Bob chooses a secret integer k_2 and sends a^{k_2} **mod** p to Alice.

(4) Alice computes $(a^{k_2})^{k_1}$ **mod** p.

(5) Bob computes $(a^{k_1})^{k_2}$ **mod** p.

At the end of this protocol, Alice and Bob have computed their shared key, namely

$$(a^{k_2})^{k_1} \textbf{ mod } p = (a^{k_1})^{k_2} \textbf{ mod } p.$$

DIFFIE-HELLMAN KEY AGREEMENT PROTOCOL

- No other method is known for finding the shared key using just the public information.
  - It is thought to be computationally infeasible when p and a are sufficiently large.
  - With the computing power available now, this system is considered unbreakable when p has more than 300 decimal digits and k1 and k2 have more than 100 decimal digits each

EXTRA ADVANCED TOPICS IN MODULAR ARITHMETIC

- The Chinese Remainder Theorem
- Computer Arithmetic with Large Integers
- Fermat's Little Theorem
- Pseudoprimes

Representations of Integers

- In the modern world, we use *decimal,* or *base* 10, *notation* to represent integers. For example when we write 965, we mean $9 \cdot 10^2 + 6 \cdot 10^1 + 5 \cdot 10^0$.
- We can represent numbers using any base b, where b is a positive integer greater than 1.
- The bases $b = 2$ (*binary*), $b = 8$ (*octal*) , and $b = 16$ (*hexadecimal*) are important for computing and communications
- The ancient Mayans used base 20 and the ancient Babylonians used base 60.

Base *b* Representations

- We can use positive integer b greater than 1 as a base

Theorem 1: Let b be a positive integer greater than 1. Then if n is a positive integer, it can be expressed uniquely in the form:

$$n = a_k b^k + a_{k-1} b^{k-1} + \ldots + a_1 b + a_0$$

where k is a nonnegative integer, $a_0, a_1, \ldots a_k$ are nonnegative integers less than b, and $a_k \neq 0$. The a_j, $j = 0, \ldots, k$ are called the base-b digits of the representation.

- The representation of n given in **Theorem 1** is called the ***base b expansion of n*** and is denoted by $(a_k a_{k-1} \ldots a_1 a_0)_b$.
- We usually omit the subscript 10 for base 10 expansions.

Binary Expansions

Most computers represent integers and do arithmetic with binary (base 2) expansions of integers. In these expansions, the only digits used are 0 and 1.

Example: What is the decimal expansion of the integer that has $(1\,0101\,1111)_2$ as its binary expansion?

Solution:

$$(1\,0101\,1111)_2 = 1\cdot 2^8 + 0\cdot 2^7 + 1\cdot 2^6 + 0\cdot 2^5 + 1\cdot 2^4 + 1\cdot 2^3 + 1\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 = 351.$$

Example: What is the decimal expansion of the integer that has $(11011)_2$ as its binary expansion?

Solution: $(11011)_2 = 1\cdot 2^4 + 1\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 = 27$

Octal Expansions

The octal expansion (base 8) uses the digits {0,1,2,3,4,5,6,7}.

Example: What is the decimal expansion of the number with octal expansion $(7016)_8$?

Solution: $7 \cdot 8^3 + 0 \cdot 8^2 + 1 \cdot 8^1 + 6 \cdot 8^0 = 3598$

Example: What is the decimal expansion of the number with octal expansion $(111)_8$?

Solution: $1 \cdot 8^2 + 1 \cdot 8^1 + 1 \cdot 8^0 = 64 + 8 + 1 = 73$

Hexadecimal Expansions

- The hexadecimal expansion uses 16 digits: {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}.
  - The letters A through F represent the decimal numbers 10 through 15.

Example: What is the decimal expansion of the number with hexadecimal expansion $(2AE0B)_{16}$?

Solution:

$$2\cdot16^4 + 10\cdot16^3 + 14\cdot16^2 + 0\cdot16^1 + 11\cdot16^0 = 175627$$

Example: What is the decimal expansion of the number with hexadecimal expansion $(E5)_{16}$?

Solution: $14\cdot16^1 + 5\cdot16^0 = 224 + 5 = 229$

Base Conversion

To construct the base b expansion of an integer n:

- Divide n by b to obtain a quotient and remainder.

  $n = bq_0 + a_0 \quad 0 \le a_0 \le b$

- The remainder, a_0, is the rightmost digit in the base b expansion of n. Next, divide q_0 by b.

  $q_0 = bq_1 + a_1 \quad 0 \le a_1 \le b$

- The remainder, a_1, is the second digit from the right in the base b expansion of n.
- Continue by successively dividing the quotients by b, obtaining the additional base b digits as the remainder. The process terminates when the quotient is 0.

Base Conversion

Example: Find the octal expansion of $(12345)_{10}$

Solution: Successively dividing by 8 gives:

- $12345 = 8 \cdot 1543 + 1$
- $1543 = 8 \cdot 192 + 7$
- $192 = 8 \cdot 24 + 0$
- $24 = 8 \cdot 3 + 0$
- $3 = 8 \cdot 0 + 3$

The remainders are the digits from right to left yielding $(30071)_8$.

ALGORITHMS FOR INTEGER OPERATIONS

ALGORITHM 2 Addition of Integers.

procedure *add*(a, b: positive integers)
{the binary expansions of a and b are $(a_{n-1}a_{n-2}\ldots a_1a_0)_2$
and $(b_{n-1}b_{n-2}\ldots b_1b_0)_2$, respectively}
$c := 0$
for $j := 0$ **to** $n-1$
$d := \lfloor (a_j + b_j + c)/2 \rfloor$
$s_j := a_j + b_j + c - 2d$
$c := d$
$s_n := c$
return $(s_0, s_1, \ldots, s_n)$ {the binary expansion of the sum is $(s_ns_{n-1}\ldots s_0)_2$}

ALGORITHMS FOR INTEGER OPERATIONS

$$ab = a(b_0 2^0 + b_1 2^1 + \cdots + b_{n-1} 2^{n-1})$$
$$= a(b_0 2^0) + a(b_1 2^1) + \cdots + a(b_{n-1} 2^{n-1}).$$

ALGORITHM 3 Multiplication of Integers.

procedure *multiply*(a, b: positive integers)
{the binary expansions of a and b are $(a_{n-1}a_{n-2}\ldots a_1a_0)_2$
and $(b_{n-1}b_{n-2}\ldots b_1b_0)_2$, respectively}
for $j := 0$ **to** $n - 1$
if $b_j = 1$ **then** $c_j :=$ a shifted j places
else $c_j := 0$
{$c_0, c_1, \ldots, c_{n-1}$ are the partial products}
$p := 0$
for $j := 0$ **to** $n - 1$
$p := p + c_j$
return p {p is the value of ab}

ALGORITHMS FOR INTEGER OPERATIONS

$$b^n = b^{a_{k-1}\cdot 2^{k-1}+\cdots+a_1\cdot 2+a_0} = b^{a_{k-1}\cdot 2^{k-1}} \cdots b^{a_1\cdot 2} \cdot b^{a_0}.$$

ALGORITHM 5 Modular Exponentiation.

procedure *modular exponentiation*(b: integer, $n = (a_{k-1}a_{k-2}\ldots a_1a_0)_2$,
m: positive integers)
$x := 1$
$power := b$ **mod** m
for $i := 0$ **to** $k - 1$
  if $a_i = 1$ **then** $x := (x \cdot power)$ **mod** m
  $power := (power \cdot power)$ **mod** m
return $x\{x$ equals b^n **mod** $m\}$

MODULAR EXPONENTIATION: EXAMPLE

EXAMPLE 12 Use Algorithm 5 to find 3^{644} **mod** 645.

Solution: Algorithm 5 initially sets $x = 1$ and $power = 3$ **mod** $645 = 3$. In the computation of 3^{644} **mod** 645, this algorithm determines 3^{2^j} **mod** 645 for $j = 1, 2, \ldots, 9$ by successively squaring and reducing modulo 645. If $a_j = 1$ (where a_j is the bit in the jth position in the binary expansion of 644, which is $(1010000100)_2$), it multiplies the current value of x by 3^{2^j} **mod** 645 and reduces the result modulo 645. Here are the steps used:

$i = 0$: Because $a_0 = 0$, we have $x = 1$ and $power = 3^2$ **mod** $645 = 9$ **mod** $645 = 9$;

$i = 1$: Because $a_1 = 0$, we have $x = 1$ and $power = 9^2$ **mod** $645 = 81$ **mod** $645 = 81$;

$i = 2$: Because $a_2 = 1$, we have $x = 1 \cdot 81$ **mod** $645 = 81$ and $power = 81^2$ **mod** $645 = 6561$ **mod** $645 = 111$;

$i = 3$: Because $a_3 = 0$, we have $x = 81$ and $power = 111^2$ **mod** $645 = 12{,}321$ **mod** $645 = 66$;

$i = 4$: Because $a_4 = 0$, we have $x = 81$ and $power = 66^2$ **mod** $645 = 4356$ **mod** $645 = 486$;

$i = 5$: Because $a_5 = 0$, we have $x = 81$ and $power = 486^2$ **mod** $645 = 236{,}196$ **mod** $645 = 126$;

$i = 6$: Because $a_6 = 0$, we have $x = 81$ and $power = 126^2$ **mod** $645 = 15{,}876$ **mod** $645 = 396$;

$i = 7$: Because $a_7 = 1$, we find that $x = (81 \cdot 396)$ **mod** $645 = 471$ and $power = 396^2$ **mod** $645 = 156{,}816$ **mod** $645 = 81$;

$i = 8$: Because $a_8 = 0$, we have $x = 471$ and $power = 81^2$ **mod** $645 = 6561$ **mod** $645 = 111$;

$i = 9$: Because $a_9 = 1$, we find that $x = (471 \cdot 111)$ **mod** $645 = 36$.