Number Theory and Cryptography

Chapter 4

With Question/Answer Animations

Chapter Summary

4.1 Divisibility and Modular Arithmetic

4.2 Integer Representations and Algorithms

4.3 Primes and Greatest Common Divisors

4.4 Solving Congruences

4.5 Applications of Congruences

4.6 Cryptography

Applications of Congruences

Section 4.5

Section Summary

- Hashing Functions
- Pseudorandom Numbers
- Check Digits

Hashing Functions

Def: A *hashing function* h assigns memory location $h(k)$ to the key k

- A common hashing function is $h(k) = k$ **mod** m, where m is # of memory locs.
    - Because this hashing function is onto, all memory locations are possible.

Example: $h(k) = k$ **mod** 111 assigns social security number to memory locations.
Some examples:

h(064212848) = 064212848 **mod** 111 = 14

h(037149212) = 037149212 **mod** 111 = 65

h(107405723) = 107405723 **mod** 111 = 14, but since 14 is already occupied, ssn is assigned to the next available position, which is 15.

- $h(k)$ is not 1-1 as there are many more possible keys than memory locations.
- A *collision* occurs when more than one record is assigned to same location,.
- Here a collision has been resolved by assigning to the first free location,

  $h(k,i) = (h(k) + i)$ **mod** m, where i runs from 0 to $m - 1$

  This is an example of a *linear probing function*.
- There are other methods of handling collisions.

Pseudorandom Numbers

- Random #s are used for many purposes, e.g., computer simulations.
- *Pseudorandom #s* are not truly random since they are generated by systematic methods.
- The *linear congruential method* is one commonly used procedure
- Four integers needed:
  - *modulus m*,
  - *multiplier a*, $2 \leq a < m$
  - *increment c*, $0 \leq c < m$
  - *seed* x_0, $0 \leq x_0 < m$.
- We generate a sequence of pseudorandom #s $\{x_n\}$, $0 \leq x_n < m \; \forall n$, by successively using the recursively defined function

$$x_{n+1} = (ax_n + c) \textbf{ mod } m.$$

(*an example of a recursive definition, discussed in Section* 5.3)

- If pseudorandom numbers between 0 and 1 are needed, then the generated numbers are divided by the modulus, x_n / m.

Pseudorandom Numbers

Ex: Find pseudorandom #s using $m = 9$, $a = 7$, $c = 4$, $x_0 = 3$.

Solution: $x_{n+1} = (7x_n + 4) \textbf{ mod } 9$, with $x_0 = 3$.

$x_1 = 7x_0 + 4 \textbf{ mod } 9 = 7\cdot 3 + 4 \textbf{ mod } 9 = 25 \textbf{ mod } 9 = 7,$

$x_2 = 7x_1 + 4 \textbf{ mod } 9 = 7\cdot 7 + 4 \textbf{ mod } 9 = 53 \textbf{ mod } 9 = 8,$

$x_3 = 7x_2 + 4 \textbf{ mod } 9 = 7\cdot 8 + 4 \textbf{ mod } 9 = 60 \textbf{ mod } 9 = 6,$

$x_4 = 7x_3 + 4 \textbf{ mod } 9 = 7\cdot 6 + 4 \textbf{ mod } 9 = 46 \textbf{ mod } 9 = 1,$

$x_5 = 7x_4 + 4 \textbf{ mod } 9 = 7\cdot 1 + 4 \textbf{ mod } 9 = 11 \textbf{ mod } 9 = 2,$

$x_6 = 7x_5 + 4 \textbf{ mod } 9 = 7\cdot 2 + 4 \textbf{ mod } 9 = 18 \textbf{ mod } 9 = 0,$

$x_7 = 7x_6 + 4 \textbf{ mod } 9 = 7\cdot 0 + 4 \textbf{ mod } 9 = 4 \textbf{ mod } 9 = 4,$

$x_8 = 7x_7 + 4 \textbf{ mod } 9 = 7\cdot 4 + 4 \textbf{ mod } 9 = 32 \textbf{ mod } 9 = 5,$

$x_9 = 7x_8 + 4 \textbf{ mod } 9 = 7\cdot 5 + 4 \textbf{ mod } 9 = 39 \textbf{ mod } 9 = 3.$

Or 3,7,8,6,1,2,0,4,5,3,7,,… repeating after generating 9 terms.

- A *pure multiplicative generator* has $c = 0$. Such a generator with modulus $2^{31} - 1$, multiplier $7^5 = 16,807$ generates $2^{31} - 2$ #s before repeating.

Check Digits: UPCs

A common method of detecting errors in strings of digits is to add an extra digit at the end, which is evaluated using a function. If the final digit is not correct, then the string provided is incorrect.

Example: Retail products are identified by their *Universal Product Codes* (*UPC*s). Usually these have 12 decimal digits, the last one being the check digit. The check digit is determined by the congruence:

$$3x_1 + x_2 + 3x_3 + x_4 + 3x_5 + x_6 + 3x_7 + x_8 + 3x_9 + x_{10} + 3x_{11} + x_{12} \equiv 0 \pmod{10}.$$

a. First 11 digits of the UPC are 79357343104. What is the check digit?

b. Is 041331021641 a valid UPC?

Solution:

a. $3{\cdot}7 + 9 + 3{\cdot}3 + 5 + 3{\cdot}7 + 3 + 3{\cdot}4 + 3 + 3{\cdot}1 + 0 + 3{\cdot}4 + x_{12} \equiv 0 \pmod{10}$

$21 + 9 + 9 + 5 + 21 + 3 + 12 + 3 + 3 + 0 + 12 + x_{12} \equiv 0 \pmod{10}$

$98 + x_{12} \equiv 0 \pmod{10}$

$x_{12} \equiv 2 \pmod{10}$ So, the check digit is 2.

b. $3{\cdot}0 + 4 + 3{\cdot}1 + 3 + 3{\cdot}3 + 1 + 3{\cdot}0 + 2 + 3{\cdot}1 + 6 + 3{\cdot}4 + 1 \equiv$

$0 + 4 + 3 + 3 + 9 + 1 + 0 + 2 + 3 + 6 + 12 + 1 = 44 \equiv 4 \not\equiv 0 \pmod{10}$

Hence, 041331021641 is not a valid UPC.

Check Digits: ISBNs

Books use the *International Standard Book Number* (ISBN-10), a 10 digit code.

- The first 9 digits identify the language, the publisher, and the book.
- The tenth digit is a check digit, which is determined by $x_{10} \equiv \sum_{i=1}^{9} i x_i \pmod{11}$.
- An ISBN-10 # is valid provided $\sum_{i=1}^{10} i x_i \equiv 0 \pmod{11}$.

X is used for the digit 10.

Example:

a. If first 9 digits of the ISBN-10 are 007288008, what is check digit?
b. Is 084930149X a valid ISBN10?

Solution:

a. $X_{10} \equiv 1\cdot0 + 2\cdot0 + 3\cdot7 + 4\cdot2 + 5\cdot8 + 6\cdot8 + 7\cdot0 + 8\cdot0 + 9\cdot8 \pmod{11}$.
$X_{10} \equiv 0 + 0 + 21 + 8 + 40 + 48 + 0 + 0 + 72 \pmod{11}$.
$X_{10} \equiv 189 \equiv 2 \pmod{11}$. Hence, $X_{10} = 2$.

b. $1\cdot0 + 2\cdot8 + 3\cdot4 + 4\cdot9 + 5\cdot3 + 6\cdot0 + 7\cdot1 + 8\cdot4 + 9\cdot9 + 10\cdot10 =$
$0 + 16 + 12 + 36 + 15 + 0 + 7 + 32 + 81 + 100 = 299 \equiv 2 \not\equiv 0 \pmod{11}$
Hence, 084930149X is not a valid ISBN-10.

- A *single error* is an error in one digit of an identification number.
- A *transposition error* is the accidental interchanging of two digits.
- Both of these error types can be detected by the ISBN and UPC schemes.