CS228

Nathan Sprague

January 29, 2014

Modular Arithmetic (Review from 4.1)

- If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$:

$$a + c \equiv b + d \pmod{m}$$

$$ac \equiv bd \pmod{m}$$

- $(a + b) \bmod m = [(a \bmod m) + (b \bmod m)] \bmod m$
- $ab \bmod m = [(a \bmod m)(b \bmod m)] \bmod m$
- Examples: $9 + 12 \pmod 5$ and $9 * 12 \pmod 5$

Hashing Functions

- $h(k)$ assigns a memory location to key k.
- Simplest:

$$h(k) = k \bmod m$$

- Where m is the number of memory locations.

Example: 8 slots, keys = { 47, 81, 32, 74, 21, 10 }

- Linear Probing:
  - $h(k, i) = (h(k) + i) \bmod m$
  - Where i is the number of collisions.

Pseudorandom Numbers

- Linear congruential method:
  - modulus m, multiplier a, increment c, seed x_0
  - $x_{n+1} = (ax_n + c) \bmod m$
  - Where:

    $2 \leq a < m$

    $0 \leq c < m$

    $0 \leq x_0 < m$

Example:

$m = 9,\ a = 7,\ c = 4,\ x_0 = 3$

Period is only 9: $\{7, 8, 6, 1, 2, 0, 4, 5, 3\}$

Check Digits - Bit Strings

Parity: $x_{n+1} = (x_1 + x_2 + \cdots + x_n) \bmod 2$
(0 if even number of ones, 1 otherwise)

Example: $01100101 \Rightarrow 0+1+1+0+0+1+0 \equiv 1 \pmod 2$
weakness: only detects odd number of errors

Check Digits - UPC Symbols

UPCs: Universal Product Codes (12 digits)
first digit = category, next five = manufacturer, next five = product, last = check digit

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

Example: 041331021641

Check Digits - ISBN

ISBNs: International Standard Book Number (10 digits) blocks for language, publisher, book number, check digit (or X which means 10)

$$x_{10} \equiv \sum_{i=1}^{9} i \cdot x_i \pmod{11}$$

Or in other words:

$$\sum_{i=1}^{10} i \cdot x_i \equiv 0 \pmod{11}$$

Our textbook: 0-07-338309-0
3*7 + 4*3 + 5*3 + 6*8 + 7*3 + 9*9
= 198
= 18 * 11