Section 2.6 – Applications of Congruences

Hashing Functions

Definition

A *hashing function* h assigns memory location $h(k)$ to the record that has k as its key.

A common hashing function is $h(k) = k \textbf{ mod } m$, where m is the number of memory locations. Because this hashing function is onto, all memory locations are possible.

Example

Find the memory locations assigned by the hashing function $h(k) = k \textbf{ mod } 111$ to the records of customers with Social Security numbers 064212848, 037149212, and 107405723.

Solution

This hashing function assigns the records of customers with social security numbers as keys to memory locations in the following manner:

$h(064212848) = 064212848 \textbf{ mod } 111 = 14$
$h(037149212) = 037149212 \textbf{ mod } 111 = 65$
$h(107405723) = 107405723 \textbf{ mod } 111 = 14,$

But since location 14 is already occupied, the record is assigned to the next available position, which is 15.

The hashing function is not one-to-one as there are many more possible keys than memory locations. When more than one record is assigned to the same location, we say a *collision* occurs. Here a collision has been resolved by assigning the record to the first free location. For collision resolution, we can use a *linear probing function*:

$$h(k,\ i) = \left(h(k),\ i\right) \textbf{ mod } m, \text{ where } i \text{ from 0 to m} - 1.$$

There are many other methods of handling with collisions. You may cover these in a later CS course.

Pseudorandom Numbers

Randomly chosen numbers are needed for many purposes, including computer simulations.
Pseudorandom numbers are not truly random since they are generated by systematic methods.
The ***linear congruential*** *method* is one commonly used procedure for generating pseudorandom numbers.
Four integers are needed: the *modulus m*, the *multiplier a*, the *increment c*, and *seed* x_0, with $2 \le a < m$, $0 \le c < m$, $0 \le x_0 < m$. We generate a sequence of pseudorandom numbers $\{x_m\}$, with $0 \le x_n < m$ for all n, by successively using the recursively defined function

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

Example

Find the sequence of pseudorandom numbers generated by the linear congruential method with modulus $m = 9$, multiplier $a = 7$, increment $c = 4$, and seed $x_0 = 3$.

Solution

Compute the terms of the sequence by successively using the congruence $x_{n+1} = \left(7x_n + 4\right) \textbf{ mod } 9$, with $x_0 = 3$.

$$x_1 = \left(7x_0 + 4\right) \textbf{ mod } 9 = \left(7\cdot 3 + 4\right) \textbf{ mod } 9 = 25 \textbf{ mod } 9 = 7$$

$$x_2 = \left(7x_1 + 4\right) \textbf{ mod } 9 = \left(7\cdot 7 + 4\right) \textbf{ mod } 9 = 53 \textbf{ mod } 9 = 8$$

$$x_3 = \left(7x_2 + 4\right) \textbf{ mod } 9 = \left(7\cdot 8 + 4\right) \textbf{ mod } 9 = 60 \textbf{ mod } 9 = 6$$

$$x_4 = \left(7x_3 + 4\right) \textbf{ mod } 9 = \left(7\cdot 6 + 4\right) \textbf{ mod } 9 = 46 \textbf{ mod } 9 = 1$$

$$x_5 = \left(7x_4 + 4\right) \textbf{ mod } 9 = \left(7\cdot 1 + 4\right) \textbf{ mod } 9 = 11 \textbf{ mod } 9 = 2$$

$$x_6 = \left(7x_5 + 4\right) \textbf{ mod } 9 = \left(7\cdot 2 + 4\right) \textbf{ mod } 9 = 18 \textbf{ mod } 9 = 0$$

$$x_7 = \left(7x_6 + 4\right) \textbf{ mod } 9 = \left(7\cdot 0 + 4\right) \textbf{ mod } 9 = 4 \textbf{ mod } 9 = 4$$

$$x_8 = \left(7x_7 + 4\right) \textbf{ mod } 9 = \left(7\cdot 4 + 4\right) \textbf{ mod } 9 = 32 \textbf{ mod } 9 = 5$$

$$x_9 = \left(7x_8 + 4\right) \textbf{ mod } 9 = \left(7\cdot 5 + 4\right) \textbf{ mod } 9 = 39 \textbf{ mod } 9 = 3$$

The sequence generated is 3, 7, 8, 6, 1, 2, 0, 4, 5, 3, 7, 8, 6, 1, 2, 0, 4, 5, 3, … It repeats after generating 9 terms.

- Commonly, computers use a linear congruential generator with increment $c = 0$. This is called a *pure multiplicative generator*. Such a generator with modulus $2^{31} - 1$ and multiplier $7^5 = 16{,}807$ generates $2^{31} - 2$ numbers 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 is assumed not to be correct.

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:

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

a) Suppose that the first 11 digits of the UPC are 79357343104. What is the check digit?

b) Is 041331021641 a valid UPC?

Solution

a) $3\cdot 7 + 9 + 3\cdot 3 + 5 + 3\cdot 7 + 3 + 3\cdot 4 + 3 + 3\cdot 1 + 0 + 3\cdot 4 + x_{12} \equiv 0 \ (\mathbf{mod}\ 10)$

$98 + x_{12} \equiv 0 \ (\mathbf{mod}\ 10)$

$x_{12} \equiv 0 \ (\mathbf{mod}\ 10)$. So, the check digit is 2.

b) $3\cdot 0 + 4 + 3\cdot 1 + 3 + 3\cdot 3 + 1 + 3\cdot 0 + 2 + 3\cdot 1 + 6 + 3\cdot 4 + 1 \equiv 0 \ (\mathbf{mod}\ 10)$

$44 \equiv 4 \not\equiv \ (\mathbf{mod}\ 10)$

Hence, 041331021641 is not a valid UPC.

Exercises *Section* 2.6 – Applications of Congruences

1. Find the memory locations assigned by the hashing function $h(k) = k$ **mod** 97 to the records of customers with Social Security numbers?

a) 034567981 *b)* 183211232 *c)* 220195744 *d)* 987255335
e) 104578690 *f)* 432222187 *g)* 372201919 *h)* 501338753

2. A parking lot has 31 visitor spaces, numbered from 0 to 30. Visitors are assigned parking spaces using the hashing function $h(k) = k$ **mod** 31, where k is the number formed from the first three digits on a visitor's license plate.

a) Which spaces are assigned by the hashing function to cars that have these first three digits on their license plates: 317, 918, 007, 100, 111, 310

b) Describe a procedure visitors should follow to find a free parking space, when the space they are assigned is occupied.

3. Find the sequence of pseudorandom numbers generated by the linear congruential generator

a) $x_{n+1} = \left(3x_n + 2\right)$ **mod** 13 with seed $x_0 = 1$.

b) $x_{n+1} = \left(4x_n + 1\right)$ **mod** 7 with seed $x_0 = 3$.

4. Find the sequence of pseudorandom numbers generated by using the pure multiplicative generator $x_{n+1} = 3x_n$ **mod** 11 with seed $x_0 = 2$.

5. The first nine digits of the ISBN-10 of the European version of the fifth edition of this book are $0-07-119881$. What is the check digit for that book?

6. The ISBN-10 of the sixth edition of Elementary Number Theory and Its Applications is $0-321-500Q1-8$, where Q is a digit. Find the value of Q.

7. The USPS sells money orders identified by 11-digit number $x_1, x_2, \ldots, x_{11}$. The first ten digits identify the money order: x_{11} is a check digit that satisfies $x_{11} = x_1 + x_2 + \cdots + x_{10}$ **mod** 9. Find the check digit for the USPS money orders that have identification number that start with these ten digits

a) 7555618873 *b)* 6966133421 *c)* 8018927435 *d)* 3289744134
e) 74051489623 *f)* 88382013445 *g)* 56152240784 *h)* 66606631178

8. Determine which single digit errors are detected by the USPS money order code.

9. Determine which transposition errors are detected by the USPS money order code.