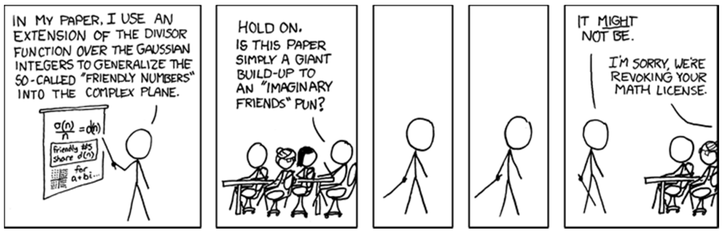
CSE 311: Foundations of Computing

Fall 2013

Lecture 11: Modular arithmetic and applications



announcements

Reading assignment

Modular arithmetic

4.1-4.3, 7th edition

3.4-3.6, 6th edition

review: divisibility

Integers a, b, with a ≠ 0, we say that a *divides* b if there is an integer k such that b = ka. The notation a | b denotes "a divides b."

review: division theorem

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

$q = a$ **div** d $\qquad$ $r = a$ **mod** d

review: modular arithmetic

Let a and b be integers, and m be a positive integer. We say a *is congruent to b modulo m* if m divides $a - b$. We use the notation $a \equiv b \pmod{m}$ to indicate that a is congruent to b modulo m.

review: modular arithmetic

Let a and b be integers, and let m be a positive integer. Then $a \equiv b \pmod{m}$ if and only if $a \bmod m = b \bmod m$.

review: modular arithmetic

Let m be a positive integer. If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then

- $a + c \equiv b + d \pmod{m}$ and
- $ac \equiv bd \pmod{m}$

example

Let n be an integer.
Prove that $n^2 \equiv 0 \pmod{4}$ or $n^2 \equiv 1 \pmod{4}$

n-bit unsigned integer representation

- Represent integer x as sum of powers of 2:
  If $x = \sum_{i=0}^{n-1} b_i 2^i$ where each $b_i \in \{0,1\}$
  then representation is $b_{n-1}...b_2\ b_1\ b_0$

  99 = 64 + 32 + 2 + 1
  18 = 16 + 2

- For n = 8:
  99: 0110 0011
  18: 0001 0010

signed integer representation

n-bit signed integers
Suppose $-2^{n-1} < x < 2^{n-1}$
First bit as the sign, n-1 bits for the value

99 = 64 + 32 + 2 + 1
18 = 16 + 2

For n = 8:
99: 0110 0011
-18: 1001 0010

Any problems with this representation?

two's complement representation

n bit signed integers, first bit will still be the sign bit

Suppose $0 \leq x < 2^{n-1}$,
x is represented by the binary representation of x
Suppose $0 \leq x \leq 2^{n-1}$,
$-x$ is represented by the binary representation of $2^n - x$

Key property: Twos complement representation of any number y is equivalent to y mod 2^n so arithmetic works mod 2^n

99 = 64 + 32 + 2 + 1
18 = 16 + 2

For n = 8:
99: 0110 0011
-18: 1110 1110

signed vs two's complement

-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
1111	1110	1101	1100	1011	1010	1001	0000	0001	0010	0011	0100	0101	0110	0111

Signed

-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
1000	1001	1010	1011	1100	1101	1110	1111	0000	0001	0010	0011	0100	0101	0110	0111

Two's complement

two's complement representation

- For $0 < x \leq 2^{n-1}$, $-x$ is represented by the binary representation of $2^n - x$

- To compute this: Flip the bits of x then add 1:
  - All 1's string is $2^n - 1$, so

    Flip the bits of x $\equiv$ replace x by $2^n - 1 - x$

basic applications of mod

- **Hashing**
- **Pseudo random number generation**
- **Simple cipher**

hashing

Scenario:

Map a small number of data values from a large domain $\{0, 1, \ldots, M-1\}$ ...

...into a small set of locations $\{0,1, \ldots, n-1\}$ so one can quickly check if some value is present

- $\text{hash}(x) = x \bmod p$ **for p a prime close to n**
  - **or** $\text{hash}(x) = (ax + b) \bmod p$

- **Depends on all of the bits of the data**
  - **helps avoid collisions due to similar values**
  - **need to manage them if they occur**

pseudo random number generation

Linear Congruential method

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

Choose random x_0, a, c, m and produce a long sequence of x_n's

simple cipher

- **Caesar cipher**, A = 1, B = 2, . . .
  - HELLO WORLD
- **Shift cipher**
  - $f(p) = (p + k) \bmod 26$
  - $f^{-1}(p) = (p - k) \bmod 26$
- **More general**
  - $f(p) = (ap + b) \bmod 26$

modular exponentiation mod 7

x	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	1	3	5
3	3	6	2	5	1	4
4	4	1	5	2	6	3
5	5	3	1	6	4	2
6	6	5	4	3	2	1

a	a^1	a^2	a^3	a^4	a^5	a^6
1						
2						
3						
4						
5						
6						

exponentiation

- **Compute** 78365^{81453}
- **Compute** $78365^{81453} \bmod 104729$
- **Output is small**
  - need to keep intermediate results small

repeated squaring – small and fast

Since $a \bmod m \equiv a \pmod m$ for any a

we have $a^2 \bmod m = (a \bmod m)^2 \bmod m$
and $a^4 \bmod m = (a^2 \bmod m)^2 \bmod m$
and $a^8 \bmod m = (a^4 \bmod m)^2 \bmod m$
and $a^{16} \bmod m = (a^8 \bmod m)^2 \bmod m$
and $a^{32} \bmod m = (a^{16} \bmod m)^2 \bmod m$

Can compute $a^k \bmod m$ **for** $k=2^i$ **in only i steps**

fast exponentiation

```
int FastExp(int a, int n, m){
   long v = (long) a;
   int exp = 1;
   for (int i = 1; i <= n; i++){
      v = (v * v) % m;
      exp = exp + exp;
      Console.WriteLine("i : " + i + ", exp : " + exp + ", v : " + v  );
   }
   return (int)v;
}
```

program trace

```
i : 1,     exp : 2,       v : 82915
i : 2,     exp : 4,       v : 95592
i : 3,     exp : 8,       v : 70252
i : 4,     exp : 16,      v : 26992
i : 5,     exp : 32,      v : 74970
i : 6,     exp : 64,      v : 71358
i : 7,     exp : 128,     v : 20594
i : 8,     exp : 256,     v : 10143
i : 9,     exp : 512,     v : 61355
i : 10,    exp : 1024,    v : 68404
i : 11,    exp : 2048,    v :  4207
i : 12,    exp : 4096,    v : 75698
i : 13,    exp : 8192,    v : 56154
i : 14,    exp : 16384,   v : 83314
i : 15,    exp : 32768,   v : 99519
i : 16,    exp : 65536,   v : 29057
```

fast exponentiation algorithm

What if the exponent is not a power of two?

$$81453 = 2^{16} + 2^{13} + 2^{12} + 2^{11} + 2^{10} + 2^{9} + 2^{5} + 2^{3} + 2^{2} + 2^{0}$$

$$a^{81453} = a^{2^{16}} \cdot a^{2^{13}} \cdot a^{2^{12}} \cdot a^{2^{11}} \cdot a^{2^{10}} \cdot a^{2^{9}} \cdot a^{2^{5}} \cdot a^{2^{3}} \cdot a^{2^{2}} \cdot a^{2^{0}}$$

$$a^{81453} \bmod m = (a^{2^{16}} \bmod m \cdot a^{2^{13}} \bmod m \cdot a^{2^{12}} \bmod m \cdot a^{2^{11}} \bmod m \cdot a^{2^{10}} \bmod m \cdot a^{2^{9}} \bmod m \cdot a^{2^{5}} \bmod m \cdot a^{2^{3}} \bmod m \cdot a^{2^{2}} \bmod m \cdot a^{2^{0}} \bmod m) \bmod m$$

The fast exponentiation algorithm computes $a^n \bmod m$ using $O(\log n)$ multiplications $\bmod m$

primality

An integer p greater than 1 is called *prime* if the only positive factors of p are 1 and p.

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

fundamental theorem of arithmetic

Every positive integer greater than 1 has a unique prime factorization

48 = 2 • 2 • 2 • 2 • 3
591 = 3 • 197
45,523 = 45,523
321,950 = 2 • 5 • 5 • 47 • 137
1,234,567,890 = 2 • 3 • 3 • 5 • 3,607 • 3,803

factorization

If n is composite, it has a factor of size at most $\sqrt{n}$.

euclid's theorem

There are an infinite number of primes.

Proof by contradiction:

Suppose that there are only a finite number of primes: $p_1, p_2, \ldots, p_n$