review: divisibility

Integers $a$, $b$, with $a \neq 0$, we say that $a$ divides $b$ if there is an integer $k$ such that $b = ka$. The notation $a \mid b$ 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 $a = dq + r$.

$q = a \div d \quad r = a \mod d$
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 $m$.

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

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:
  
  \[
  x = \sum_{i=0}^{n-1} b_i 2^i \quad \text{where each } b_i \in \{0, 1\}
  \]
  
  then representation is \( b_{n-1}...b_2 b_1 b_0 \)

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

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

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: Two’s 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

Signed


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

- hash($x$) = $x \mod p$ for $p$ a prime close to $n$
  - or hash($x$) = ($ax + b$) $\mod p$

- 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

\[ x_{n+1} = (a \cdot x_n + c) \mod m \]

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
  - \( f(p) = (p + k) \mod 26 \)
  - \( f^{-1}(p) = (p - k) \mod 26 \)

- More general
  - \( f(p) = (ap + b) \mod 26 \)

modular exponentiation mod 7

<table>
<thead>
<tr>
<th>p</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>a^1</th>
<th>a^2</th>
<th>a^3</th>
<th>a^4</th>
<th>a^5</th>
<th>a^6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

exponentiation

- Compute \( 78365^{81453} \)

- Compute \( 78365^{81453} \mod 104729 \)

- Output is small
  - need to keep intermediate results small

repeated squaring – small and fast

Since \( a \mod m \equiv a \mod m \) for any \( a \)

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

Can compute \( a^k \mod m \) for \( k=2^i \) in only \( i \) steps
**fast exponentiation**

```csharp
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?

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

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

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

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

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

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: $p_1, p_2, \ldots, p_n$