Announcements

- Reading assignments
  - Today and Monday:
    • 4.3 7th Edition
    • 3.5, 3.6 6th Edition
    • 2.5, 2.6 up to p. 191 5th Edition
  - Wednesday
    • Start on induction
- Homework 4
  - Available now (posted Wednesday night)

Highlights from last lecture

- Introduction of modular arithmetic
  What is the difference between \( r = a \mod d \) and \( r \equiv a \pmod{d} \)?

- Fumbling with the projector and whiteboard (morning lecture)

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 \( a = dq + r \).

\[
q = a \div d \quad r = a \mod d
\]

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 \( m \).

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

\[
a + c \equiv b + d \pmod{m} \quad \text{and} \quad ac \equiv bd \pmod{m}
\]
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:
  \[ x = \sum_{i=0}^{n-1} b_i 2^i \]
  where each \( b_i \in \{0,1\} \)
  then representation is \( b_{n-1} \ldots b_2 b_1 b_0 \)

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

**Key property:** Two’s complement representation of any number \( y \) is equivalent to \( y \pmod{2^n} \) so arithmetic works \( \pmod{2^n} \)

- \( 99 = 64 + 32 + 2 + 1 \)
- \( 18 = 16 + 2 \)

For \( n = 8 \):
- 99: 0110 0011
- -18: 1110 1110

### Two’s complement representation

- Suppose \( 0 < x \leq 2^{n-1} \), \( -x \) 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 \( x \) = replace \( x \) by \( 2^n-1-x \)

### Basic applications of mod

- Hashing
- Pseudo random number generation
- Simple cipher
Hashing

- Map values from a large domain, 0...M-1 in a much smaller domain, 0...n-1
- Index lookup
- Test for equality
- Hash(x) = x mod p
- Often want the hash function to depend on all of the bits of the data
  - Collision management

Pseudo Random number generation

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

Simple cipher

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

Modular Exponentiation

\[
\begin{align*}
\text{x} & \mid 1 & 2 & 3 & 4 & 5 & 6 \\
1 & 1 & 2 & 3 & 4 & 5 & 6 \\
2 & 2 & 4 & 6 & 1 & 3 & 5 \\
3 & 3 & 6 & 2 & 5 & 1 & 4 \\
4 & 4 & 1 & 5 & 2 & 6 & 3 \\
5 & 5 & 3 & 1 & 6 & 4 & 2 \\
6 & 6 & 5 & 4 & 3 & 2 & 1 \\
\end{align*}
\]

Exponentiation

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

Fast exponentiation

```csharp
int FastExp(int x, int n)
{
    long v = (long) x;
    int m = 1;
    for (int i = 1; i <= n; i++)
    {
        v = (v * v) % modulus;
        m = m + m;
        Console.WriteLine("i : "+i+", m : "+m+", v : "+v +");
    }
    return (int)v;
}
```
Program Trace

<table>
<thead>
<tr>
<th>i</th>
<th>m</th>
<th>v</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>82915</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>95592</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>70252</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>26992</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>74970</td>
</tr>
<tr>
<td>6</td>
<td>64</td>
<td>71358</td>
</tr>
<tr>
<td>7</td>
<td>128</td>
<td>20594</td>
</tr>
<tr>
<td>8</td>
<td>256</td>
<td>10143</td>
</tr>
<tr>
<td>9</td>
<td>512</td>
<td>75698</td>
</tr>
<tr>
<td>10</td>
<td>1024</td>
<td>68404</td>
</tr>
<tr>
<td>11</td>
<td>2048</td>
<td>83314</td>
</tr>
<tr>
<td>12</td>
<td>4096</td>
<td>99519</td>
</tr>
<tr>
<td>13</td>
<td>8192</td>
<td>29057</td>
</tr>
</tbody>
</table>

Fast exponentiation algorithm

- What if the exponent is not a power of two?

\[ 81453 = 2^{16} + 2^{13} + 2^{12} + 2^{11} + 2^{10} + 2^9 + 2^6 + 2^3 + 2^0 \]

The fast exponentiation algorithm computes \( a^n \mod p \) in time \( O(\log n) \)

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

- \( 48 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 3 \)
- \( 591 = 3 \cdot 107 \)
- \( 45,523 = 45,523 \)
- \( 321,850 = 2 \cdot 5 \cdot 17 \cdot 97 \cdot 313 \)
- \( 1,234,567,890 = 2 \cdot 3 \cdot 3 \cdot 5 \cdot 3,607 \cdot 3,803 \)

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 there are a finite number of primes: \( p_1, p_2, \ldots, p_n \)
Distribution of Primes

If you pick a random number \( n \) in the range \([x, 2x]\), what is the chance that \( n \) is prime?

Famous Algorithmic Problems

- Primality Testing:
  - Given an integer \( n \), determine if \( n \) is prime
- Factoring
  - Given an integer \( n \), determine the prime factorization of \( n \)

Primality Testing

- Is the following 200 digit number prime:
  
  40992408416096028179761232525875254029092850990862201340392052540955208352606215439915948260875718993797824735118621138119256949864080611330666502556080656092533901288801362035441884878187944219033