CSE 311 Section 5

Number Theory

Administrivia

Announcements & Reminders

- HW3
  - If you think something was graded incorrectly, submit a regrade request!
- HW4 was due yesterday
  - Use late days if you need them!
- HW5
  - Due next week on Oct 30th at 11:59PM on Gradescope

Greatest Common Divisor

Some Definitions

- Greatest Common Divisor (GCD):
  - The Greatest Common Divisor of a and b ($\gcd(a, b)$) is the largest integer c such that $c|a$ and $c|b$
- Multiplicative Inverse:
  - The multiplicative inverse of $a \pmod{n}$ is an integer b such that $ab \equiv 1 \pmod{n}$

Problem 1 – Warm-Up

a) Calculate gcd(100, 50).

a) Calculate gcd(17, 31)

a) Find the multiplicative inverse of 6 (mod 7).

a) Does 49 have a multiplicative inverse (mod 7)?

Try this problem with the people around you, and then we'll go over it together!

Extended Euclidean Algorithm

Finding GCD

GCD Facts:

If a and b are positive integers, then:

$$\gcd(a, b) = \gcd(b, a\%b)$$

$$\gcd(a, 0) = a$$

```
public int GCD(int m, int n){
    if(m<n){
        int temp = m;
        m=n;
        n=temp;
    }
    while(n != 0) {
        int rem = m % n;
        m=n;
        n=temp;
    }
    return m;
}
```

$$\gcd(a, b) = \gcd(b, a\%b)$$

Euclid's Algorithm

gcd(660,126)

$$\gcd(a, b) = \gcd(b, a\%b)$$

Euclid's Algorithm

gcd(660,126) = gcd(126, 660 % 126) = gcd(126, 30)

$$\gcd(a, b) = \gcd(b, a\%b)$$

Euclid's Algorithm

gcd(660,126) = gcd(126, 660 % 126) = gcd(126, 30)

= gcd(30, 126 % 30) = gcd(30, 6)

$$\gcd(a, b) = \gcd(b, a\%b)$$

Euclid's Algorithm

gcd(660,126) = gcd(126, 660 % 126) = gcd(126, 30)

= gcd(30, 126 % 30) = gcd(30, 6)

= gcd(6, 30 % 6) = gcd(6, 0)

$$\gcd(a, b) = \gcd(b, a\%b)$$

Euclid's Algorithm

gcd(660,126) = gcd(126, 660 % 126) = gcd(126, 30)
= gcd(30, 126 % 30) = gcd(30, 6)
= gcd(6, 30 % 6) = gcd(6, 0)
= 6

$$\gcd(a, b) = \gcd(b, a\%b)$$

Euclid's Algorithm

gcd(660,126) = gcd(126, 660 % 126) = gcd(126, 30)
= gcd(30, 126 % 30) = gcd(30, 6)
= gcd(6, 30 % 6) = gcd(6, 0)
= 6

Tableau form

$$660 = 5 \cdot 126 + 30$$
$$126 = 4 \cdot 30 + 6$$
$$30 = 5 \cdot 6 + 0$$

Bézout's Theorem

- Bézout's Theorem:
  - If a and b are positive integers, then there exist integers s and t such that

$$\gcd(a, b) = sa + tb$$

- We're not going to prove this theorem in section though, because it's hard and ugly

Extended Euclidean Algorithm

Bézout's Theorem tells us that $\gcd(a, b) = sa + tb$.

To find the s, t we can use the Extended Euclidean Algorithm.

- Step 1: compute $\gcd(a, b)$; keep tableau information
- Step 2: solve all equations for the remainder
- Step 3: substitute backward

Extended Euclidean Algorithm

gcd(35,27)

- **Compute $gcd(a, b)$; keep tableau information**
- Solve all equations for the remainder
- Substitute backward

Extended Euclidean Algorithm

gcd(35,27) = gcd(27, 35%27) = gcd(27,8)

- **Compute $gcd(a, b)$; keep tableau information**
- Solve all equations for the remainder
- Substitute backward

Extended Euclidean Algorithm

gcd(35,27) = gcd(27, 35%27) = gcd(27,8)
= gcd(8, 27%8) = gcd(8, 3)

- **Compute $gcd(a, b)$; keep tableau information**
- Solve all equations for the remainder
- Substitute backward

Extended Euclidean Algorithm

gcd(35,27) = gcd(27, 35%27) = gcd(27,8)
= gcd(8, 27%8) = gcd(8, 3)
= gcd(3, 8%3) = gcd(3, 2)

- **Compute $gcd(a, b)$; keep tableau information**
- Solve all equations for the remainder
- Substitute backward

Extended Euclidean Algorithm

gcd(35,27) = gcd(27, 35%27) = gcd(27,8)
= gcd(8, 27%8) = gcd(8, 3)
= gcd(3, 8%3) = gcd(3, 2)
= gcd(2, 3%2) = gcd(2,1)

- **Compute $gcd(a, b)$; keep tableau information**
- Solve all equations for the remainder
- Substitute backward

Extended Euclidean Algorithm

gcd(35,27) = gcd(27, 35%27) = gcd(27,8)
= gcd(8, 27%8) = gcd(8, 3)
= gcd(3, 8%3) = gcd(3, 2)
= gcd(2, 3%2) = gcd(2,1)
= gcd(1, 2%1) = gcd(1,0)

- **Compute $gcd(a, b)$; keep tableau information**
- Solve all equations for the remainder
- Substitute backward

Extended Euclidean Algorithm

- **Compute $gcd(a, b)$; keep tableau information**
- Solve all equations for the remainder
- Substitute backward

gcd(35,27)	= gcd(27, 35%27)	= gcd(27,8)
	= gcd(8, 27%8)	= gcd(8, 3)
	= gcd(3, 8%3)	= gcd(3, 2)
	= gcd(2, 3%2)	= gcd(2,1)
	= gcd(1, 2%1)	= gcd(1,0)

$$
\begin{aligned}
35 &= 1 \cdot 27 + 8 \\
27 &= 3 \cdot 8 + 3 \\
8 &= 2 \cdot 3 + 2 \\
3 &= 1 \cdot 2 + 1
\end{aligned}
$$

Extended Euclidean Algorithm

- Compute $gcd(a, b)$; keep tableau information
- **Solve all equations for the remainder**
- Substitute backward

$$
\begin{aligned}
35 &= 1 \cdot 27 + 8 \\
27 &= 3 \cdot 8 + 3 \\
8 &= 2 \cdot 3 + 2 \\
3 &= 1 \cdot 2 + 1
\end{aligned}
$$

Extended Euclidean Algorithm

- Compute $gcd(a, b)$; keep tableau information
- **Solve all equations for the remainder**
- Substitute backward

$$35 = 1 \cdot 27 + 8$$
$$27 = 3 \cdot 8 + 3$$
$$8 = 2 \cdot 3 + 2$$
$$3 = 1 \cdot 2 + 1$$

$$8 = 35 - 1 \cdot 27$$

Extended Euclidean Algorithm

- Compute $gcd(a, b)$; keep tableau information
- **Solve all equations for the remainder**
- Substitute backward

$$35 = 1 \cdot 27 + 8$$
$$27 = 3 \cdot 8 + 3$$
$$8 = 2 \cdot 3 + 2$$
$$3 = 1 \cdot 2 + 1$$

$$8 = 35 - 1 \cdot 27$$
$$3 = 27 - 3 \cdot 8$$

Extended Euclidean Algorithm

- Compute $gcd(a, b)$; keep tableau information
- **Solve all equations for the remainder**
- Substitute backward

$$
\begin{aligned}
35 &= 1 \cdot 27 + 8 \\
27 &= 3 \cdot 8 + 3 \\
8 &= 2 \cdot 3 + 2 \\
3 &= 1 \cdot 2 + 1
\end{aligned}
$$

$$
\begin{aligned}
8 &= 35 - 1 \cdot 27 \\
3 &= 27 - 3 \cdot 8 \\
2 &= 8 - 2 \cdot 3
\end{aligned}
$$

Extended Euclidean Algorithm

- Compute $gcd(a, b)$; keep tableau information
- **Solve all equations for the remainder**
- Substitute backward

$$
\begin{aligned}
35 &= 1 \cdot 27 + 8 \\
27 &= 3 \cdot 8 + 3 \\
8 &= 2 \cdot 3 + 2 \\
3 &= 1 \cdot 2 + 1
\end{aligned}
$$

$$
\begin{aligned}
8 &= 35 - 1 \cdot 27 \\
3 &= 27 - 3 \cdot 8 \\
2 &= 8 - 2 \cdot 3 \\
1 &= 3 - 1 \cdot 2
\end{aligned}
$$

Extended Euclidean Algorithm

```
8 = 35 - 1·27
3 = 27 - 3·8
2 = 8  - 2·3
1 = 3  - 1·2
```

- Compute $gcd(a, b)$; keep tableau information
- Solve all equations for the remainder
- **Substitute backward**

Extended Euclidean Algorithm

$$8 = 35 - 1 \cdot 27$$
$$3 = 27 - 3 \cdot 8$$
$$2 = 8 - 2 \cdot 3$$
$$1 = 3 - 1 \cdot 2$$

$$1 = 3 - 1 \cdot 2$$

- Compute $gcd(a, b)$; keep tableau information
- Solve all equations for the remainder
- **Substitute backward**

Extended Euclidean Algorithm

- Compute $gcd(a, b)$; keep tableau information
- Solve all equations for the remainder
- **Substitute backward**

```
8 = 35 - 1·27
3 = 27 - 3·8
2 = 8  - 2·3
1 = 3  - 1·2
```

```
1 = 3 - 1·2
  = 3 - 1·(8 - 2·3)
```

Extended Euclidean Algorithm

- Compute $gcd(a, b)$; keep tableau information
- Solve all equations for the remainder
- **Substitute backward**

```
8 = 35 - 1·27
3 = 27 - 3·8
2 = 8  - 2·3
1 = 3  - 1·2
```

```
1 = 3 - 1·2
  = 3 - 1·(8 - 2·3)
  = -1·8 + 3·3
```

Extended Euclidean Algorithm

- Compute $gcd(a, b)$; keep tableau information
- Solve all equations for the remainder
- **Substitute backward**

```
8 = 35 - 1·27
3 = 27 - 3·8
2 = 8  - 2·3
1 = 3  - 1·2
```

```
1 = 3 - 1·2
  = 3 - 1·(8 - 2·3)
  = -1·8 + 3·3
  = -1·8 + 3(27 - 3·8)
```

Extended Euclidean Algorithm

- Compute $gcd(a, b)$; keep tableau information
- Solve all equations for the remainder
- **Substitute backward**

```
8 = 35 - 1·27
3 = 27 - 3·8
2 = 8  - 2·3
1 = 3  - 1·2
```

```
1 = 3 - 1·2
  = 3 - 1·(8 - 2·3)
  = -1·8 + 3·3
  = -1·8 + 3(27 - 3·8)
  = 3·27 - 10·8
```

Extended Euclidean Algorithm

- Compute $gcd(a, b)$; keep tableau information
- Solve all equations for the remainder
- **Substitute backward**

```
8 = 35 - 1·27
3 = 27 - 3·8
2 = 8  - 2·3
1 = 3  - 1·2
```

```
1 = 3 - 1·2
  = 3 - 1·(8 - 2·3)
  = -1·8 + 3·3
  = -1·8 + 3(27 - 3·8)
  = 3·27 - 10·8
  = 3·27 - 10(35 - 1·27)
```

Extended Euclidean Algorithm

- Compute $gcd(a, b)$; keep tableau information
- Solve all equations for the remainder
- **Substitute backward**

```
8 = 35 - 1·27
3 = 27 - 3·8
2 = 8  - 2·3
1 = 3  - 1·2
```

```
1 = 3 - 1·2
  = 3 - 1·(8 - 2·3)
  = -1·8 + 3·3
  = -1·8 + 3(27 - 3·8)
  = 3·27 - 10·8
  = 3·27 - 10(35 - 1·27)
  = 13·27 - 10·35
```

Extended Euclidean Algorithm

- Compute $gcd(a, b)$; keep tableau information
- Solve all equations for the remainder
- **Substitute backward**

$$
\begin{aligned}
8 &= 35 - 1 \cdot 27 \\
3 &= 27 - 3 \cdot 8 \\
2 &= 8 - 2 \cdot 3 \\
1 &= 3 - 1 \cdot 2
\end{aligned}
$$

When substituting back, you keep the larger of m, n and the number you just substituted.

Don't simplify further! (or you'll lose the form you need)

$$
\begin{aligned}
1 &= 3 - 1 \cdot 2 \\
&= 3 - 1 \cdot (8 - 2 \cdot 3) \\
&= -1 \cdot 8 + 3 \cdot 3 \\
&= -1 \cdot 8 + 3(27 - 3 \cdot 8) \\
&= 3 \cdot 27 - 10 \cdot 8 \\
&= 3 \cdot 27 - 10(35 - 1 \cdot 27) \\
&= 13 \cdot 27 - 10 \cdot 35
\end{aligned}
$$

Problem 2 – Extended Euclidean Algorithm

a) Find the multiplicative inverse y of $7 \bmod 33$. That is, find y such that $7y \equiv 1 \ (mod \ 33)$. You should use the extended Euclidean Algorithm. Your answer should be in the range $0 \leq y < 33$.

b) Now, solve $7z \equiv 2 \ (\mathrm{mod}\ 33)$ for all of its integer solutions z.

Try this problem with the people around you, and then we'll go over it together!

Proof by Contradiction

5b: argue code is valid using contradiction

Problem 5b - Proof by Contradiction

```
public boolean isPrime(int n) {
    int potentialDiv = 2;
    while (potentialDiv < n) {
        if (n % potentialDiv == 0)
            return false;
        potentialDiv++;
    }
    return true;
}
```

(Given)

Returns true if and only if *n* is prime (assume n > 0).

Problem 5b - Proof by Contradiction

```
public boolean isPrime(int n) {
    int potentialDiv = 2;
    while (potentialDiv <= Math.sqrt(n)) {
        if (n % potentialDiv == 0)
            return false;
        potentialDiv++;
    }
    return true;
}
```

(Modified)

(*Maybe*) Returns true if and only if *n* is prime (assume n > 0).

Number Theory

Bonus! :D

Some Definitions

- Divides:
  - For $a, b \in \mathbb{Z}$: $a \mid b$ iff $\exists (k \in \mathbb{Z})\ b = ka$
  - For integers a and b, we say a divides b if and only if there exists an integer k such that $b = ka$
- Congruence Modulo:
  - For $a, b \in \mathbb{Z}, m \in \mathbb{Z}^+$: $a \equiv b \pmod{m}$ iff $m \mid (b - a)$
  - For integers a and b and positive integer m, we say a is congruent to b modulo m if and only if m divides $b - a$

Problem 4 – Modular Arithmetic

a) Prove that if $a \mid b$ and $b \mid a$, where a and b are integers, then $a = b$ or $a = -b$.

b) Prove that if $n \mid m$, where n and m are integers greater than 1, and if $a \equiv b \pmod{m}$, where a and b are integers, then $a \equiv b \pmod{n}$.

Lets walk through part (a) together.

Problem 4 – Modular Arithmetic

a) Prove that if $a \mid b$ and $b \mid a$, where a and b are integers, then $a = b$ or $a = -b$.

b) Prove that if $n \mid m$, where n and m are integers greater than 1, and if $a \equiv b \pmod{m}$, where a and b are integers, then $a \equiv b \pmod{n}$.

Now try part (b) with the people around you, and then we'll go over it together!

That's All, Folks!

Thanks for coming to section this week!
Any questions?