

# MA/CSSE 473

## Day 08

**Euclid's Algorithm**

**Modular Division**



# MA/CSSE 473 Day 08

- HW 3 is due now, HW 4 is available.
- Why was Homework 2 average score so low?
  - Partial answer: three students did not turn it in; one did not get to the last four problems.
- **Student Questions on homework, other things.**
- Euclid's algorithm: analysis
- Extended Euclid Algorithm
- Modular Division
- Look at HW4



# Recap: Euclid's Algorithm for gcd

```
def euclid(a, b):  
    """ INPUT:  Two integers a and b with a >= b >= 0  
        OUTPUT: gcd(a, b) """  
    if b == 0:  
        return a  
    return euclid(b, a % b)
```

- Example: euclid(60, 36)
- Does the algorithm work?
- How efficient is it?



# Euclid's Algorithm: the analysis

```
def euclid(a, b):  
    """ INPUT: Two integers a and b with a >= b >= 0  
        OUTPUT: gcd(a, b) """  
    if b == 0:  
        return a  
    return euclid(b, a % b)
```

- Lemma: If  $a \geq b$ , then  $a \bmod b < a/2$
- Proof
  - If  $b \leq a/2$ , then  $a \bmod b < b \leq a/2$
  - If  $b > a/2$ , then  $a \bmod b = a - b < a/2$
- Application
  - After two recursive calls, both  $a$  and  $b$  are at most half of what they were, (i.e. reduced by 1 bit)
  - Thus if  $a$  and  $b$  have  $n$  bits, at most  $2n$  recursive calls are needed.
  - Each recursive call involves a division,  $\Theta(n^2)$
  - Entire algorithm is  $\Theta(n^3)$



# gcd and linear combinations

- Lemma: If  $d$  divides both  $a$  and  $b$ , and  $d = ax + by$  for some integers  $x$  and  $y$ , then  $d = \gcd(a, b)$
- Proof
  - By the first of the two conditions,  $d$  is a common divisor of  $a$  and  $b$ . It cannot exceed the greatest common divisor, so  $d \leq \gcd(a, b)$
  - $\gcd(a, b)$  is a common divisor of  $a$  and  $b$ , so it must divide  $ax + by = d$ . Thus  $\gcd(a, b) \leq d$
  - Putting these together,  $\gcd(a, b) = d$
- If we can supply the  $x$  and  $y$  as in the lemma, we know that  $d$  is the gcd.
- It turns out that a simple modification of Euclid's algorithm will calculate the  $x$  and  $y$ .



# Extended Euclid Algorithm

```
def euclidExtended(a, b):  
    """ INPUT: Two integers a and b with  $a \geq b \geq 0$   
        OUTPUT: Integers x, y, d such that  $d = \text{gcd}(a, b)$   
            and  $d = ax + by$  """  
    print "    ", a, b  
    if b == 0:  
        return 1, 0, a  
    x, y, d = euclidExtended(b, a % b)  
    return y, x - a/b*y, d
```

- Proof that it works

- First, the number d it produces really is the gcd of a and b.
- We can just ignore the x and y values, and we have the same algorithm as before.



# Extended Euclid Algorithm: proof

```
def euclidExtended(a, b):  
    """ INPUT: Two integers a and b with  $a \geq b \geq 0$   
        OUTPUT: Integers x, y, d such that  $d = \text{gcd}(a, b)$   
            and  $d = ax + by$  """  
    print "      ", a, b  
    if b == 0:  
        return 1, 0, a  
    x, y, d = euclidExtended(b, a % b)  
    return y, x - a/b*y, d
```

- Proof that it works

- First, the number d it produces really is  $\text{gcd}(a, b)$ 
  - We can just ignore the x and y values, and we have the same algorithm as before.
- We must show that the x and y it returns are such that  $ax + by = d$ .
- We do that by induction on b.



# Proof that $ax+by = d$ (induction on $b$ )

- **Base case:  $b=0$**

Then  $\gcd(a,b) = a$ ,  
and the algorithm  
produces  $x = 1$ ,  
 $y = 0$ . ✓

```
def euclidExtended(a, b):  
    """ INPUT: Two integers a and b with a >= b >  
        OUTPUT: Integers x, y, d such that d = gcd  
            and d = ax + by"""  
    print "      ", a, b  
    if b == 0:  
        return 1, 0, a  
    x, y, d = euclidExtended(b, a % b)  
    return y, x - a/b*y, d
```

- **Induction step:  $b > 0$ .**

- It finds  $\gcd(a, b)$  by calling  $\text{euclidExtended}(b, a\%b)$
- Since  $a\%b$  is smaller than  $b$ , by induction the  $x'$  and  $y'$  returned by the recursive call are such that

$$\gcd(b, a \% b) = bx' + (a \% b)y'$$

- We can write  $a \% b$  as  $a - \lfloor a/b \rfloor * b$
- $d = \gcd(a, b) = \gcd(b, a\%b) = bx' + (a\%b)y'$   
 $= bx' + (a - \lfloor a/b \rfloor * b)y' = ay' + b(x' - \lfloor a/b \rfloor y')$
- Thus  $x = y'$  and  $y = x' - \lfloor a/b \rfloor y'$  are the numbers that make  $ax + by = d$
- This  $x$  and  $y$  are the numbers returned by the algorithm.



# Example: gcd (25, 11)

- $25 = 2 * 11 + 3$
- $11 = 3 * 3 + 2$
- $3 = 1 * 2 + 1$
- $2 = 2 * 1 + 0$ , so  $\text{gcd}(11, 25) = 1$ .
- **Now work backwards**
- $1 = 1 - 0$ . Substitute  $0 = 2 - 2 * 1$
- $1 = 1 - (2 - 2 * 1) = -1 * 2 + 3 * 1$ . Substitute  $1 = 3 - 1 * 2$
- $1 = -1 * 2 + 3(3 - 1 * 2) = 3 * 3 - 4 * 2$ . Substitute  $2 = 11 - 3 * 3$
- $1 = 3 * 3 - 4 * (11 - 3 * 3) = -4 * 11 + 15 * 3$ . Substitute  $3 = 25 - 2 * 11$
- $1 = -4(11) + 15(25 - 2 * 11) = -34 * 11 + 15 * 25$
- Thus  $x = 15$  and  $y = -34$  **Done!**



# Trominoes Homework (HW 04)

- Can be done with another person (turn in one write-up together) if you wish.
- Some parts are routine, others require creativity.

