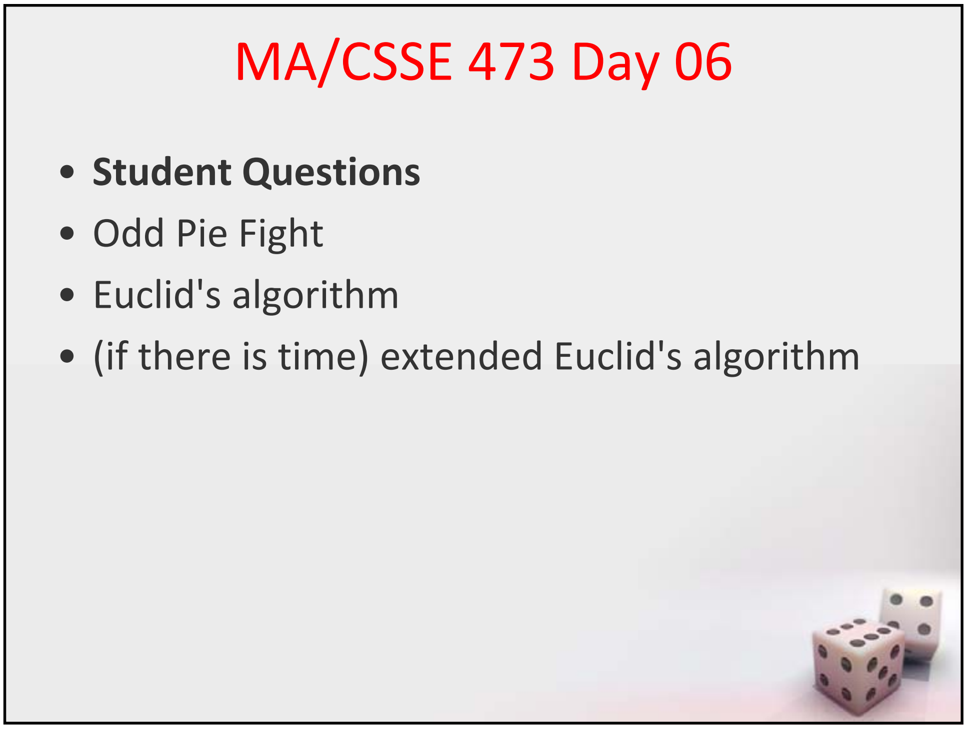
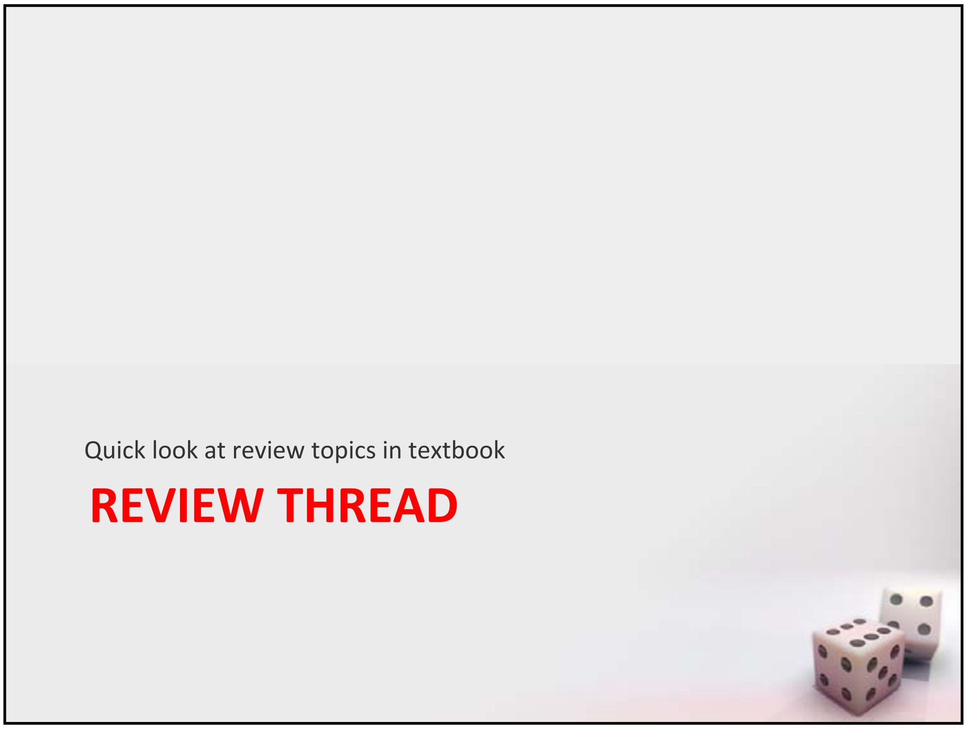
MA/CSSE 473 Day 06

Euclid's Algorithm



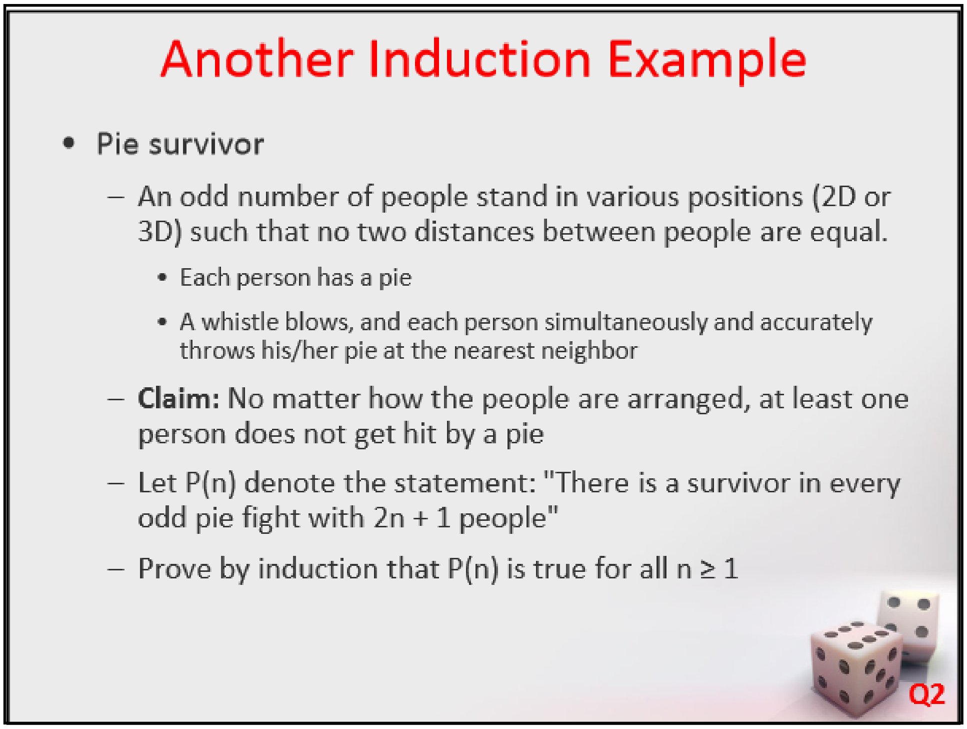
MA/CSSE 473 Day 06

- **Student Questions**
- Odd Pie Fight
- Euclid's algorithm
- (if there is time) extended Euclid's algorithm



Quick look at review topics in textbook

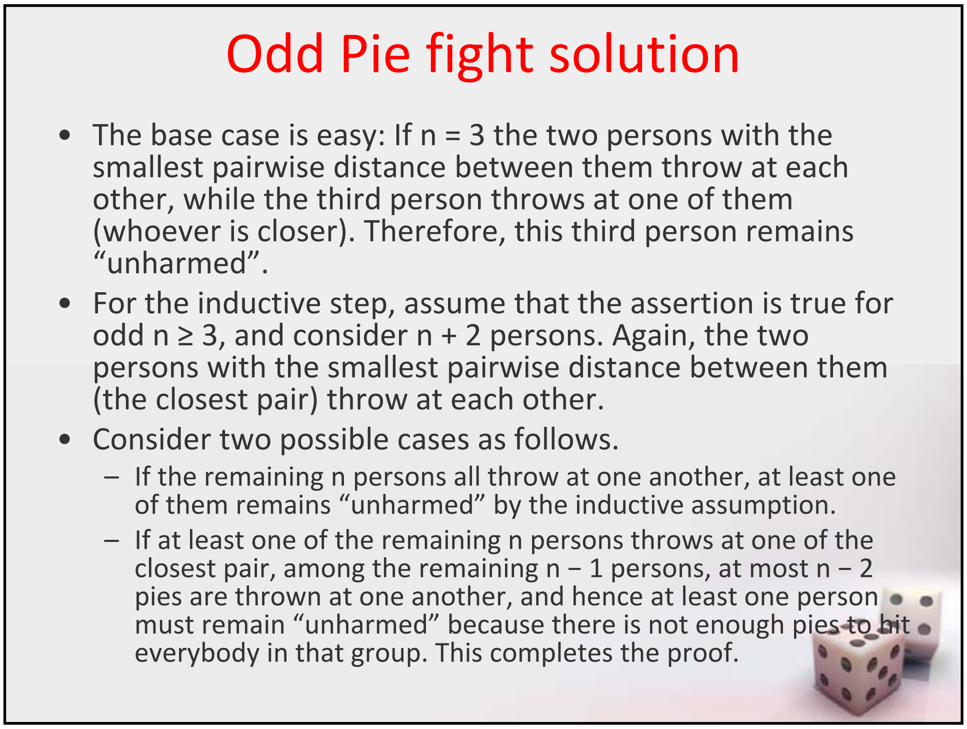
REVIEW THREAD



Another Induction Example

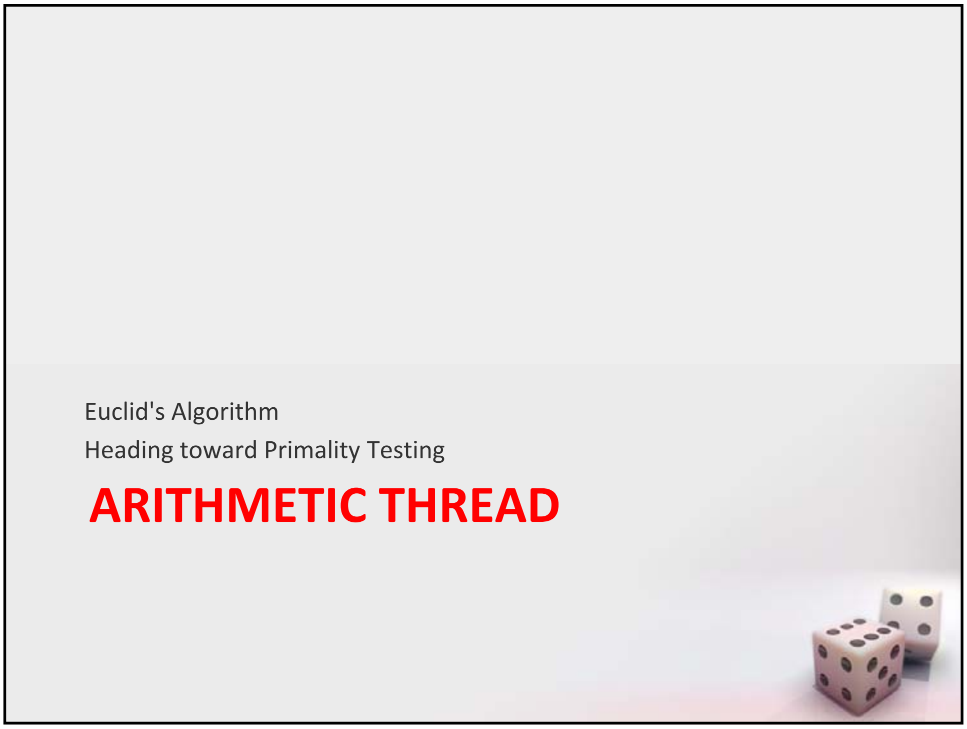
- Pie survivor
  - An odd number of people stand in various positions (2D or 3D) such that no two distances between people are equal.
    - Each person has a pie
    - A whistle blows, and each person simultaneously and accurately throws his/her pie at the nearest neighbor
  - **Claim:** No matter how the people are arranged, at least one person does not get hit by a pie
  - Let P(n) denote the statement: "There is a survivor in every odd pie fight with 2n + 1 people"
  - Prove by induction that P(n) is true for all n ≥ 1

Q2



Odd Pie fight solution

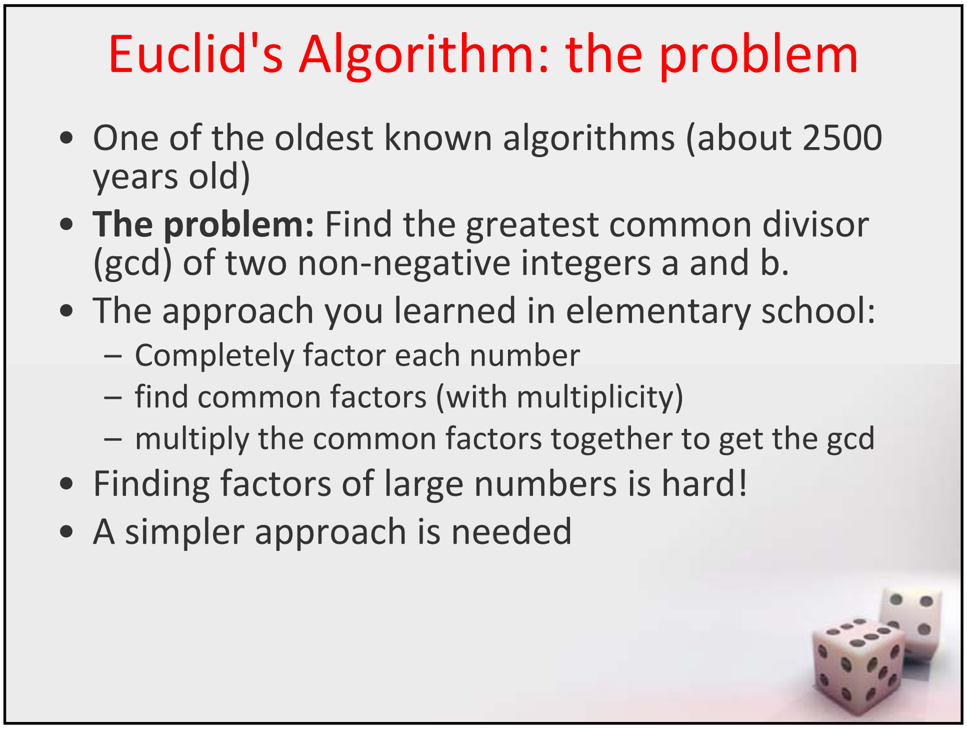
- The base case is easy: If n = 3 the two persons with the smallest pairwise distance between them throw at each other, while the third person throws at one of them (whoever is closer). Therefore, this third person remains “unharmed”.
- For the inductive step, assume that the assertion is true for odd n ≥ 3, and consider n + 2 persons. Again, the two persons with the smallest pairwise distance between them (the closest pair) throw at each other.
- Consider two possible cases as follows.
  - If the remaining n persons all throw at one another, at least one of them remains “unharmed” by the inductive assumption.
  - If at least one of the remaining n persons throws at one of the closest pair, among the remaining n − 1 persons, at most n − 2 pies are thrown at one another, and hence at least one person must remain “unharmed” because there is not enough pies to hit everybody in that group. This completes the proof.



Euclid's Algorithm

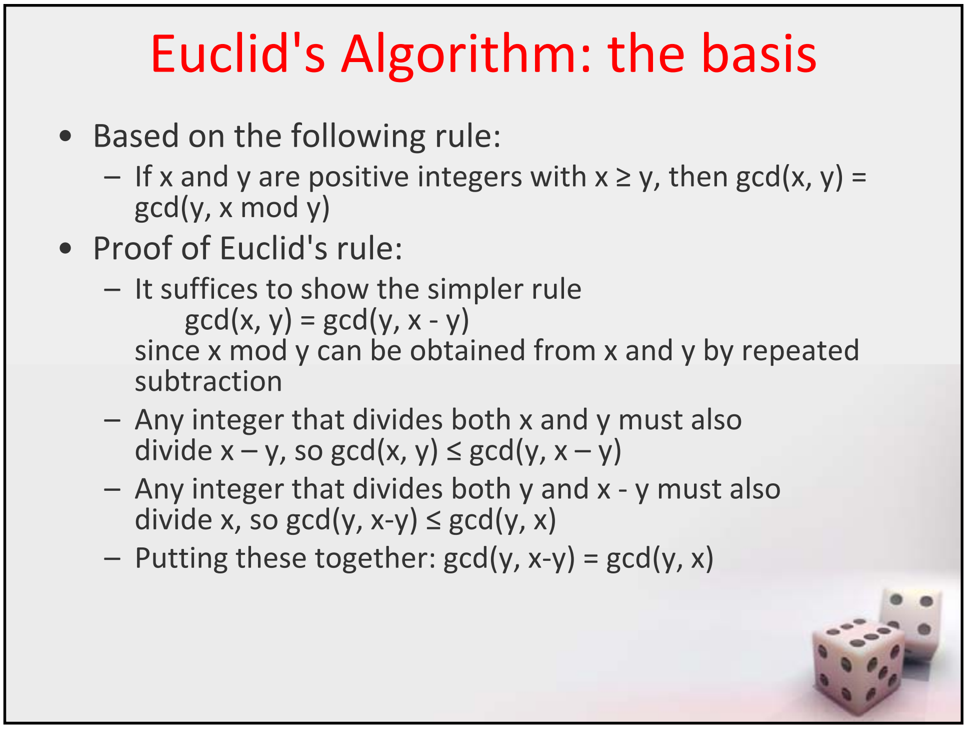
Heading toward Primality Testing

ARITHMETIC THREAD



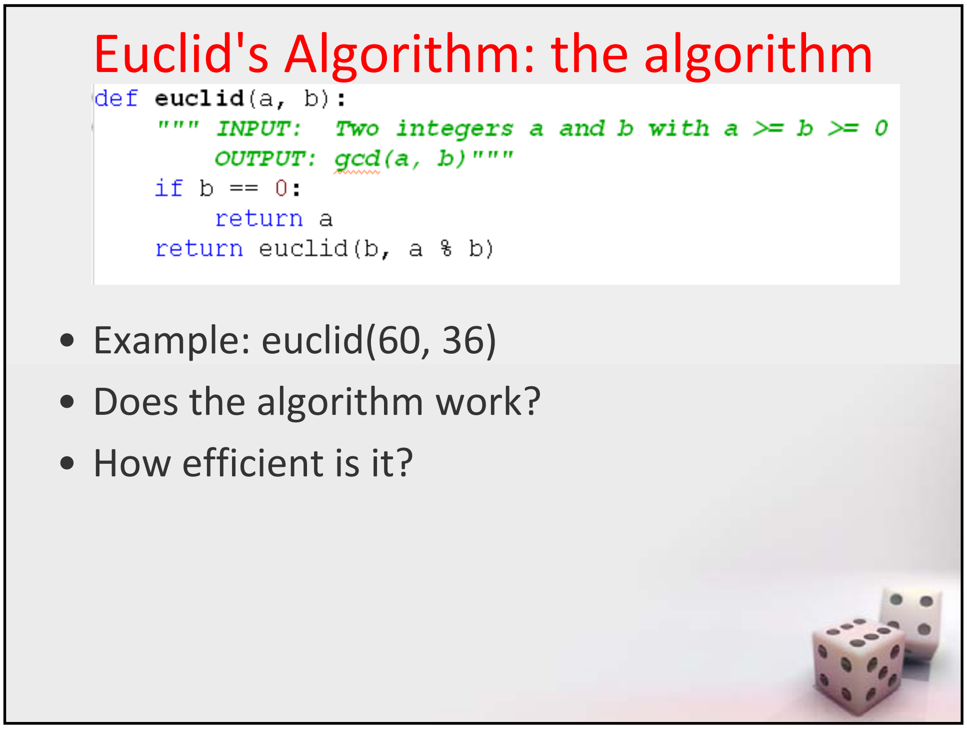
Euclid's Algorithm: the problem

- One of the oldest known algorithms (about 2500 years old)
- **The problem:** Find the greatest common divisor (gcd) of two non-negative integers a and b.
- The approach you learned in elementary school:
  - Completely factor each number
  - find common factors (with multiplicity)
  - multiply the common factors together to get the gcd
- Finding factors of large numbers is hard!
- A simpler approach is needed



Euclid's Algorithm: the basis

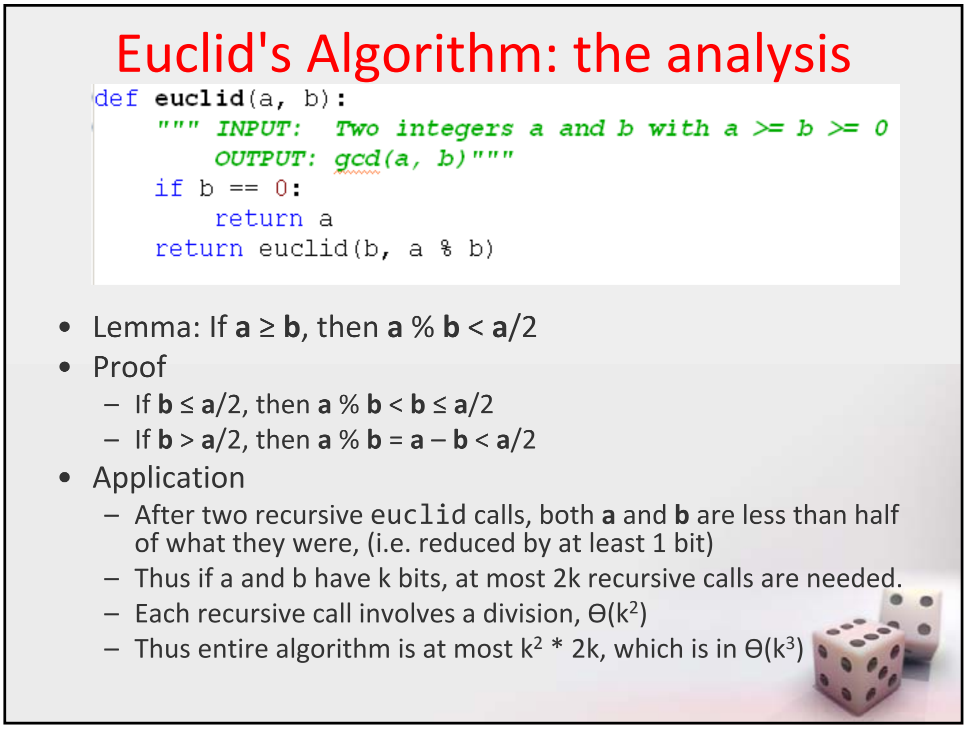
- Based on the following rule:
  - If x and y are positive integers with $x \geq y$, then $\gcd(x, y) = \gcd(y, x \bmod y)$
- Proof of Euclid's rule:
  - It suffices to show the simpler rule
    $\gcd(x, y) = \gcd(y, x - y)$
    since x mod y can be obtained from x and y by repeated subtraction
  - Any integer that divides both x and y must also divide $x – y$, so $\gcd(x, y) \leq \gcd(y, x – y)$
  - Any integer that divides both y and $x - y$ must also divide x, so $\gcd(y, x-y) \leq \gcd(y, x)$
  - Putting these together: $\gcd(y, x-y) = \gcd(y, x)$



Euclid's Algorithm: the algorithm

```
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)
```

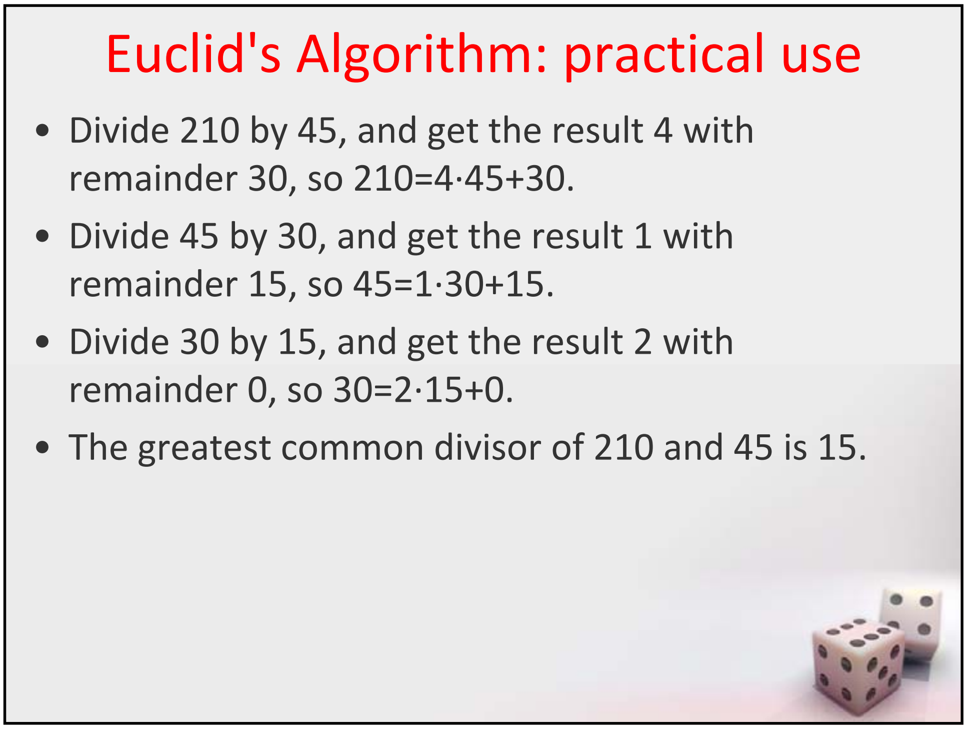
- Example: euclid(60, 36)
- Does the algorithm work?
- How efficient is it?



Euclid's Algorithm: the analysis

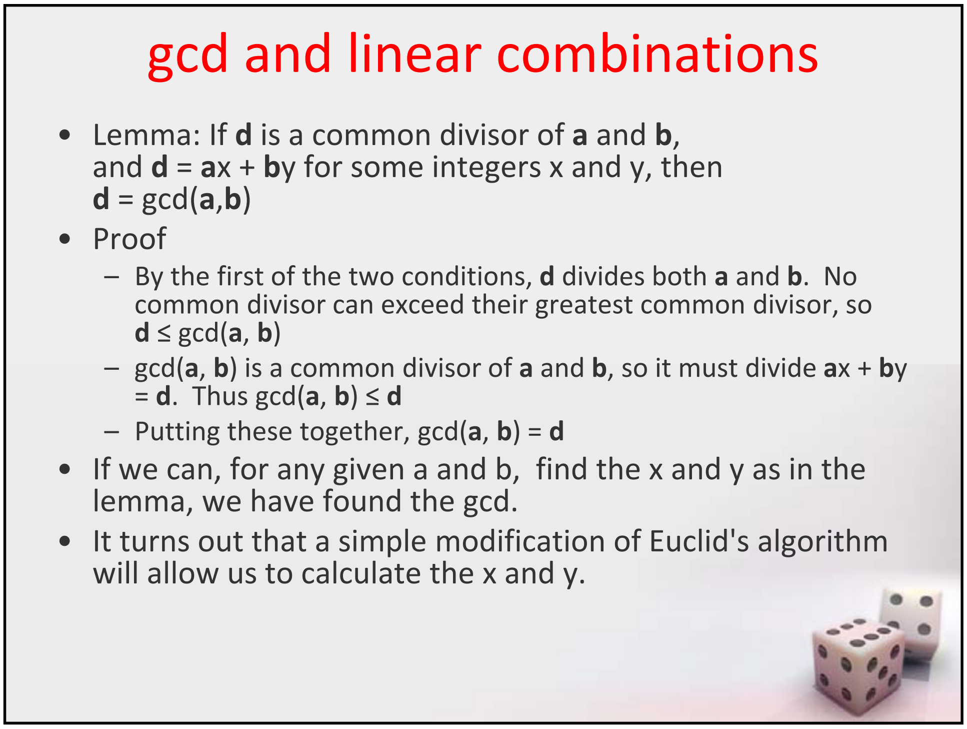
```
def euclid(a, b):
    """ INPUT:  Two integers a and b with a >= b >= 0
        OUTPUT: gcd(a, b)"""
    if b == 0:
        return a
    return euclid(b, a % b)
```

- Lemma: If **a** ≥ **b**, then **a** % **b** < **a**/2
- Proof
  - If **b** ≤ **a**/2, then **a** % **b** < **b** ≤ **a**/2
  - If **b** > **a**/2, then **a** % **b** = **a** – **b** < **a**/2
- Application
  - After two recursive `euclid` calls, both **a** and **b** are less than half of what they were, (i.e. reduced by at least 1 bit)
  - Thus if a and b have k bits, at most 2k recursive calls are needed.
  - Each recursive call involves a division, $\Theta(k^2)$
  - Thus entire algorithm is at most k^2 * 2k, which is in $\Theta(k^3)$



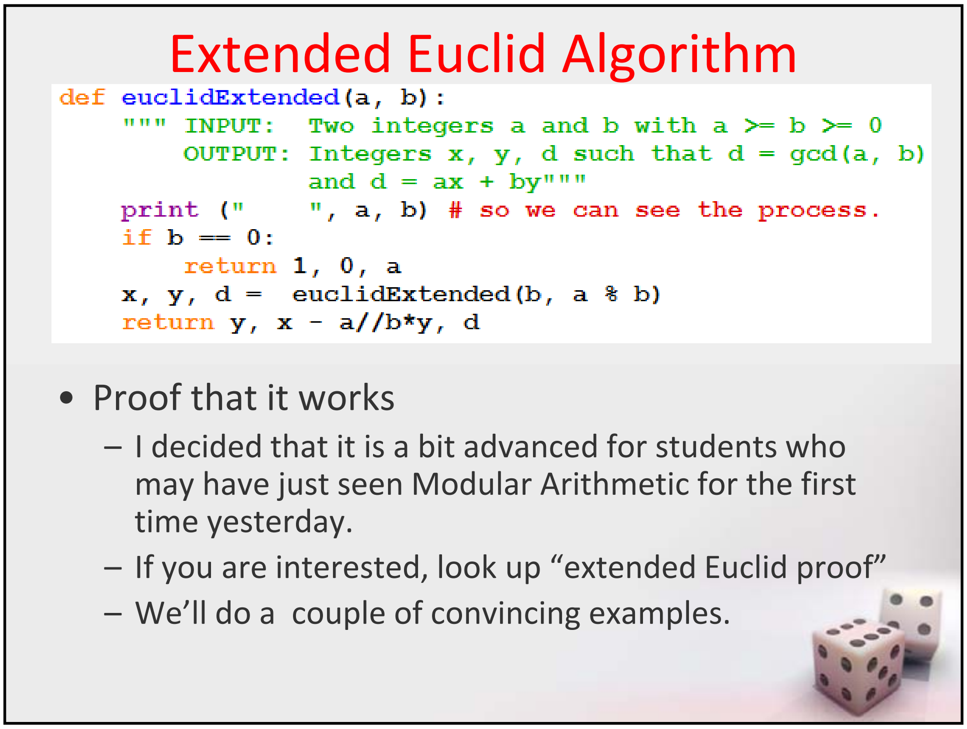
Euclid's Algorithm: practical use

- Divide 210 by 45, and get the result 4 with remainder 30, so $210=4\cdot45+30$.
- Divide 45 by 30, and get the result 1 with remainder 15, so $45=1\cdot30+15$.
- Divide 30 by 15, and get the result 2 with remainder 0, so $30=2\cdot15+0$.
- The greatest common divisor of 210 and 45 is 15.



gcd and linear combinations

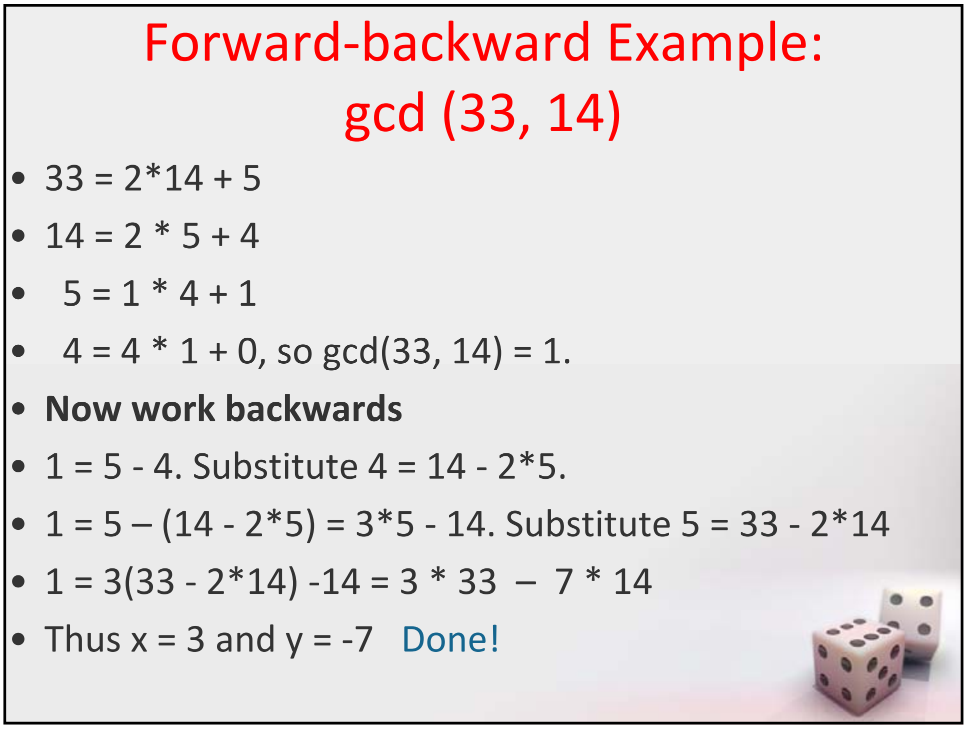
- Lemma: If **d** is a common divisor of **a** and **b**, and **d** = **a**x + **b**y for some integers x and y, then **d** = gcd(**a**,**b**)
- Proof
  - By the first of the two conditions, **d** divides both **a** and **b**. No common divisor can exceed their greatest common divisor, so **d** ≤ gcd(**a**, **b**)
  - gcd(**a**, **b**) is a common divisor of **a** and **b**, so it must divide **a**x + **b**y = **d**. Thus gcd(**a**, **b**) ≤ **d**
  - Putting these together, gcd(**a**, **b**) = **d**
- If we can, for any given a and b, find the x and y as in the lemma, we have found the gcd.
- It turns out that a simple modification of Euclid's algorithm will allow us to calculate the x and y.



Extended Euclid Algorithm

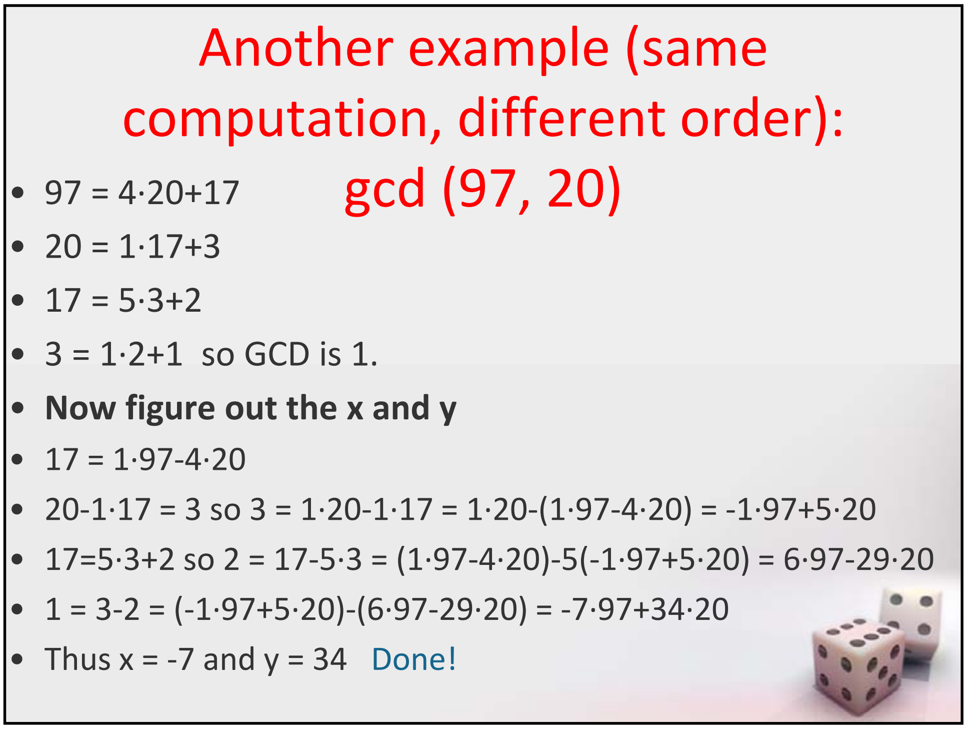
```
def euclidExtended(a, b):
    """ INPUT:  Two integers a and b with a >= b >= 0
        OUTPUT: Integers x, y, d such that d = gcd(a, b)
                and d = ax + by"""
    print ("    ", a, b) # so we can see the process.
    if b == 0:
        return 1, 0, a
    x, y, d =  euclidExtended(b,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

- Proof that it works
  - I decided that it is a bit advanced for students who may have just seen Modular Arithmetic for the first time yesterday.
  - If you are interested, look up “extended Euclid proof”
  - We’ll do a couple of convincing examples.



Forward-backward Example: gcd (33, 14)

- 33 = 2*14 + 5
- 14 = 2 * 5 + 4
- 5 = 1 * 4 + 1
- 4 = 4 * 1 + 0, so gcd(33, 14) = 1.
- **Now work backwards**
- 1 = 5 - 4. Substitute 4 = 14 - 2*5.
- 1 = 5 – (14 - 2*5) = 3*5 - 14. Substitute 5 = 33 - 2*14
- 1 = 3(33 - 2*14) -14 = 3 * 33 – 7 * 14
- Thus x = 3 and y = -7   Done!



Another example (same computation, different order): gcd (97, 20)

- $97 = 4\cdot 20+17$
- $20 = 1\cdot 17+3$
- $17 = 5\cdot 3+2$
- $3 = 1\cdot 2+1$ so GCD is 1.
- **Now figure out the x and y**
- $17 = 1\cdot 97-4\cdot 20$
- $20-1\cdot 17 = 3$ so $3 = 1\cdot 20-1\cdot 17 = 1\cdot 20-(1\cdot 97-4\cdot 20) = -1\cdot 97+5\cdot 20$
- $17=5\cdot 3+2$ so $2 = 17-5\cdot 3 = (1\cdot 97-4\cdot 20)-5(-1\cdot 97+5\cdot 20) = 6\cdot 97-29\cdot 20$
- $1 = 3-2 = (-1\cdot 97+5\cdot 20)-(6\cdot 97-29\cdot 20) = -7\cdot 97+34\cdot 20$
- Thus $x = -7$ and $y = 34$ Done!