

## Questions/Complaints About Homework?

Here's the procedure for homework questions/complaints:

1. Read the solutions first.
2. Talk to the person who graded it (check initials)
3. If (1) and (2) don't work, talk to me.

Further comments:

- There's no statute of limitations on grade changes
  - although asking questions right away is a good strategy
- Remember that 10/12 homeworks count. Each one is roughly worth 50 points, and homework is 35% of your final grade.
  - 16 homework points = 1% on your final grade
- Remember we're grading about 80 homeworks and graders are not expected to be mind readers. It's **your** problem to write clearly.
- Don't forget to staple your homework pages together, add the cover sheet, and put your name on clearly.
  - I'll deduct 2 points if that's not the case

1

## Division

For  $a, b \in \mathbb{Z}$ ,  $a \neq 0$ ,  $a$  divides  $b$  if there is some  $c \in \mathbb{Z}$  such that  $b = ac$ .

- Notation:  $a \mid b$
- Examples:  $3 \mid 9$ ,  $3 \nmid 7$

If  $a \mid b$ , then  $a$  is a *factor* of  $b$ ,  $b$  is a *multiple* of  $a$ .

**Theorem 1:** If  $a, b, c \in \mathbb{Z}$ , then

1. if  $a \mid b$  and  $a \mid c$  then  $a \mid (b + c)$ .
2. If  $a \mid b$  then  $a \mid (bc)$
3. If  $a \mid b$  and  $b \mid c$  then  $a \mid c$  (divisibility is transitive).

**Proof:** How do you prove this? Use the definition!

- E.g., if  $a \mid b$  and  $a \mid c$ , then, for some  $d_1$  and  $d_2$ ,
$$b = ad_1 \text{ and } c = ad_2.$$
- That means  $b + c = a(d_1 + d_2)$
- So  $a \mid (b + c)$ .

Other parts: homework.

**Corollary 1:** If  $a \mid b$  and  $a \mid c$ , then  $a \mid (mb + nc)$  for any integers  $m$  and  $n$ .

3

## Algorithmic number theory

Number theory used to be viewed as the purest branch of pure mathematics.

- Now it's the basis for most modern cryptography.
- Absolutely critical for e-commerce
  - How do you know your credit card number is safe?

Goal:

- To give you a basic understanding of the mathematics behind the RSA cryptosystem
  - Need to understand how prime numbers work

2

## The division algorithm

**Theorem 2:** For  $a \in \mathbb{Z}$  and  $d \in \mathbb{N}$ ,  $d > 0$ , there exist unique  $q, r \in \mathbb{Z}$  such that  $a = q \cdot d + r$  and  $0 \leq r < d$ .

- $r$  is the remainder when  $a$  is divided by  $d$

**Notation:**  $r \equiv a \pmod{d}$ ;  $a \bmod d = r$

**Examples:**

- Dividing 101 by 11 gives a quotient of 9 and a remainder of 2 ( $101 \equiv 2 \pmod{11}$ ;  $101 \bmod 11 = 2$ ).
- Dividing 18 by 6 gives a quotient of 3 and a remainder of 0 ( $18 \equiv 0 \pmod{6}$ ;  $18 \bmod 6 = 0$ ).

**Proof:** Let  $q = \lfloor a/d \rfloor$  and define  $r = a - q \cdot d$ .

- So  $a = q \cdot d + r$  with  $q \in \mathbb{Z}$  and  $0 \leq r < d$  (since  $q \cdot d \leq a$ ).

But why are  $q$  and  $d$  unique?

- Suppose  $q \cdot d + r = q' \cdot d + r'$  with  $q', r' \in \mathbb{Z}$  and  $0 \leq r' < d$ .
- Then  $(q' - q)d = (r - r')$  with  $-d < r - r' < d$ .
- The lhs is divisible by  $d$  so  $r = r'$  and we're done.

4

## Primes

- If  $p \in N$ ,  $p > 1$  is *prime* if its only positive factors are 1 and  $p$ .
- $n \in N$  is *composite* if  $n > 1$  and  $n$  is not prime.
  - If  $n$  is composite then  $a \mid n$  for some  $a \in N$  with  $1 < a < n$
  - Can assume that  $a \leq \sqrt{n}$ .
    - \* **Proof:** By contradiction:  
Suppose  $n = bc$ ,  $b > \sqrt{n}$ ,  $c > \sqrt{n}$ . But then  $bc > n$ , a contradiction.

Primes: 2, 3, 5, 7, 11, 13, ...

Composites: 4, 6, 8, 9, ...

5

## The Fundamental Theorem of Arithmetic

**Theorem 3:** Every natural number  $n > 1$  can be uniquely represented as a product of primes, written in nondecreasing size.

- Examples:  $54 = 2 \cdot 3^3$ ,  $100 = 2^2 \cdot 5^2$ ,  $15 = 3 \cdot 5$ .

Proving that that  $n$  can be written as a product of primes is easy (by strong induction):

- Base case: 2 is the product of primes (just 2)
- Inductive step: If  $n > 2$  is prime, we are done. If not,  $n = ab$ .
  - Must have  $a < n$ ,  $b < n$ .
  - By I.H., both  $a$  and  $b$  can be written as a product of primes
  - So  $n$  is product of primes

Proving uniqueness is harder.

- We'll do that in a few days ...

7

## Primality testing

How can we tell if  $n \in N$  is prime?

The naive approach: check if  $k \mid n$  for every  $1 < k < n$ .

- But at least  $10^{m-1}$  numbers are  $\leq n$ , if  $n$  has  $m$  digits
  - 1000 numbers less than 1000 (a 4-digit number)
  - 1,000,000 less than 1,000,000 (a 7-digit number)

So the algorithm is *exponential time!*

We can do a little better

- Skip the even numbers
- That saves a factor of 2  $\rightarrow$  not good enough
- Try only primes (Sieve of Eratosthenes)
  - Still doesn't help much

We can do much better:

- There is a polynomial time *randomized* algorithm
  - We will discuss this when we talk about probability
- In 2002, Agarwal, Saxena, and Kayal gave a (non-probabilistic) polynomial time algorithm
  - Saxena and Kayal were undergrads in 2002!

6

## An Algorithm for Prime Factorization

**Fact:** If  $a$  is the smallest number  $> 1$  that divides  $n$ , then  $a$  is prime.

**Proof:** By contradiction. (Left to the reader.)

- A *multiset* is like a set, except repetitions are allowed
  - $\{\{2, 2, 3, 3, 5\}\}$  is a multiset, not a set

### PF( $n$ ): A prime factorization procedure

**Input:**  $n \in N^+$

**Output:** PFS - a multiset of  $n$ 's prime factors

PFS :=  $\emptyset$

**for**  $a = 2$  to  $\lfloor \sqrt{n} \rfloor$  **do**

**if**  $a \mid n$  **then** PFS := PF( $n/a$ )  $\cup$   $\{a\}$  **return** PFS

**if** PFS =  $\emptyset$  **then** PFS :=  $\{n\}$  [ $n$  is prime]

**Example:** PF(7007) =  $\{\{7\}\} \cup$  PF(1001)  
 =  $\{\{7, 7\}\} \cup$  PF(143)  
 =  $\{\{7, 7, 11\}\} \cup$  PF(13)  
 =  $\{\{7, 7, 11, 13\}\}$ .

8

## The Complexity of Factoring

Algorithm PF runs in exponential time:

- We're checking every number up to  $\sqrt{n}$

Can we do better?

- We don't know.
- Modern-day cryptography implicitly depends on the fact that we can't!

9

## How Many Primes Are There?

**Theorem 4:** [Euclid] There are infinitely many primes.

**Proof:** By contradiction.

- Suppose that there are only finitely many primes:  $p_1, \dots, p_n$ .
- Consider  $q = p_1 \times \dots \times p_n + 1$
- Clearly  $q > p_1, \dots, p_n$ , so it can't be prime.
- So  $q$  must have a prime factor, which must be one of  $p_1, \dots, p_n$  (since these are the only primes).
- Suppose it is  $p_i$ .
  - Then  $p_i \mid q$  and  $p_i \mid p_1 \times \dots \times p_n$
  - So  $p_i \mid (q - p_1 \times \dots \times p_n)$ ; i.e.,  $p_i \mid 1$  (Corollary 1)
  - Contradiction!

Largest currently-known prime (as of 5/04):

- $2^{24036583} - 1$ : 7235733 digits
- Check [www.utm.edu/research/primes](http://www.utm.edu/research/primes)

Primes of the form  $2^p - 1$  where  $p$  is prime are called *Mersenne primes*.

- Search for large primes focuses on Mersenne primes

10

## The distribution of primes

There are quite a few primes out there:

- Roughly one in every  $\log(n)$  numbers is prime

Formally: let  $\pi(n)$  be the number of primes  $\leq n$ :

**Prime Number Theorem:**  $\pi(n) \sim n/\log(n)$ ; that is,

$$\lim_{n \rightarrow \infty} \pi(n)/(n/\log(n)) = 1$$

Why is this important?

- Cryptosystems like RSA use a secret key that is the product of two large (100-digit) primes.
- How do you find two large primes?
  - Roughly one of every 100 100-digit numbers is prime
  - To find a 100-digit prime;
    - \* Keep choosing odd numbers at random
    - \* Check if they are prime (using fast randomized primality test)
    - \* Keep trying until you find one
    - \* Roughly 100 attempts should do it

11

## (Some) Open Problems Involving Primes

- Are there infinitely many Mersenne primes?
- *Goldbach's Conjecture*: every even number greater than 2 is the sum of two primes.
  - E.g.,  $6 = 3 + 3$ ,  $20 = 17 + 3$ ,  $28 = 17 + 11$
  - This has been checked out to  $6 \times 10^{16}$  (as of 2003)
  - Every sufficiently large integer ( $> 10^{43,000!}$ ) is the sum of four primes
- Two prime numbers that differ by two are *twin primes*
  - E.g.: (3,5), (5,7), (11,13), (17,19), (41,43)
  - also 4, 648, 619, 711,  $505 \times 2^{60,000} \pm 1!$

Are there infinitely many twin primes?

All these conjectures are believed to be true, but no one has proved them.

12

## Greatest Common Divisor (gcd)

**Definition:** For  $a \in \mathbb{Z}$  let  $D(a) = \{k \in \mathbb{N} : k \mid a\}$

- $D(a) = \{\text{divisors of } a\}$ .

**Claim.**  $|D(a)| < \infty$  if (and only if)  $a \neq 0$ .

**Proof:** If  $a \neq 0$  and  $k \mid a$ , then  $0 < k < a$ .

**Definition:** For  $a, b \in \mathbb{Z}$ ,  $CD(a, b) = D(a) \cap D(b)$  is the set of common divisors of  $a, b$ .

**Definition:** The *greatest common divisor* of  $a$  and  $b$  is

$$\gcd(a, b) = \max(CD(a, b)).$$

**Examples:**

- $\gcd(6, 9) = 3$
- $\gcd(13, 100) = 1$
- $\gcd(6, 45) = 3$

**Def.**  $a$  and  $b$  are *relatively prime* if  $\gcd(a, b) = 1$ .

- **Example:** 4 and 9 are relatively prime.
- Two numbers are relatively prime iff they have no common prime factors.

Efficient computation of  $\gcd(a, b)$  lies at the heart of commercial cryptography.

13

## Least Common Multiple (lcm)

**Definition:** The *least common multiple* of  $a, b \in \mathbb{N}^+$ ,  $\text{lcm}(a, b)$ , is the smallest  $n \in \mathbb{N}^+$  such that  $a \mid n$  and  $b \mid n$ .

- **Examples:**  $\text{lcm}(4, 9) = 36$ ,  $\text{lcm}(4, 10) = 20$ .

14

## Computing the GCD

There is a method for calculating the gcd that goes back to Euclid:

- **Recall:** if  $n > m$  and  $q$  divides both  $n$  and  $m$ , then  $q$  divides  $n - m$  and  $n + m$ .

Therefore  $\gcd(n, m) = \gcd(m, n - m)$ .

- **Proof:** Show that  $CD(n, m) = CD(m, n - m)$ ; i.e. show that  $q$  divides both  $n$  and  $m$  iff  $q$  divides both  $m$  and  $n - m$ . (If  $q$  divides  $n$  and  $m$ , then  $q$  divides  $n - m$  by the argument above. If  $q$  divides  $m$  and  $n - m$ , then  $q$  divides  $m + (n - m) = n$ .)
- This allows us to reduce the gcd computation to a simpler case.

We can do even better:

- $\gcd(n, m) = \gcd(m, n - m) = \gcd(m, n - 2m) = \dots$
- keep going as long as  $n - qm \geq 0$  —  $\lfloor n/m \rfloor$  steps

Consider  $\gcd(6, 45)$ :

- $\lfloor 45/6 \rfloor = 7$ ; remainder is 3 ( $45 \equiv 3 \pmod{6}$ )
- $\gcd(6, 45) = \gcd(6, 45 - 7 \times 6) = \gcd(6, 3) = 3$

15

We can keep this up this procedure to compute  $\gcd(n_1, n_2)$ :

- If  $n_1 \geq n_2$ , write  $n_1$  as  $q_1 n_2 + r_1$ , where  $0 \leq r_1 < n_2$ 
  - $q_1 = \lfloor n_1/n_2 \rfloor$
- $\gcd(n_1, n_2) = \gcd(r_1, n_2)$
- Now  $r_1 < n_2$ , so switch their roles:
- $n_2 = q_2 r_1 + r_2$ , where  $0 \leq r_2 < r_1$
- $\gcd(r_1, n_2) = \gcd(r_1, r_2)$
- Notice that  $\max(n_1, n_2) > \max(r_1, n_2) > \max(r_1, r_2)$
- Keep going until we have a remainder of 0 (i.e., something of the form  $\gcd(r_k, 0)$  or  $(\gcd(0, r_k))$ )
  - This is bound to happen sooner or later

16

## Euclid's Algorithm

**Input**  $m, n$  [ $m, n$  natural numbers,  $m \geq n$ ]  
 $num \leftarrow m; denom \leftarrow n$  [Initialize  $num$  and  $denom$ ]  
**repeat until**  $denom = 0$   
     $q \leftarrow \lfloor num/denom \rfloor$   
     $rem \leftarrow num - (q * denom)$  [ $num \bmod denom = rem$ ]  
     $num \leftarrow denom$  [New  $num$ ]  
     $denom \leftarrow rem$  [New  $denom$ ; note  $num \geq denom$ ]  
**endrepeat**  
**Output**  $num$  [ $num = \gcd(m, n)$ ]

Example:  $\gcd(84, 33)$

Iteration 1:  $num = 84, denom = 33, q = 2, rem = 18$

Iteration 2:  $num = 33, denom = 18, q = 1, rem = 15$

Iteration 3:  $num = 18, denom = 15, q = 1, rem = 3$

Iteration 4:  $num = 15, denom = 3, q = 5, rem = 0$

Iteration 5:  $num = 3, denom = 0 \Rightarrow \gcd(84, 33) = 3$

17

## Euclid's Algorithm: Complexity

**Input**  $m, n$  [ $m, n$  natural numbers,  $m \geq n$ ]  
 $num \leftarrow m; denom \leftarrow n$  [Initialize  $num$  and  $denom$ ]  
**repeat until**  $denom = 0$   
     $q \leftarrow \lfloor num/denom \rfloor$   
     $rem \leftarrow num - (q * denom)$   
     $num \leftarrow denom$  [New  $num$ ]  
     $denom \leftarrow rem$  [New  $denom$ ; note  $num \geq denom$ ]  
**endrepeat**  
**Output**  $num$  [ $num = \gcd(m, n)$ ]

How many times do we go through the loop in the Euclidean algorithm:

- Best case: Easy. Never!
- Average case: Too hard
- Worst case: Can't answer this exactly, but we can get a good upper bound.
  - See how fast  $denom$  goes down in each iteration.

19

## Euclid's Algorithm: Correctness

How do we know this works?

- We need to prove that
  - (a) the algorithm terminates and
  - (b) that it correctly computes the gcd

We prove (a) and (b) simultaneously by finding appropriate loop invariants and using induction:

- Notation: Let  $num_k$  and  $denom_k$  be the values of  $num$  and  $denom$  at the beginning of the  $k$ th iteration.

$P(k)$  has three parts:

(1)  $0 < num_{k+1} + denom_{k+1} < num_k + denom_k$

(2)  $0 \leq denom_k \leq num_k$ .

(3)  $\gcd(num_k, denom_k) = \gcd(m, n)$

- Termination follows from parts (1) and (2): if  $num_k + denom_k$  decreases and  $0 \leq denom_k \leq num_k$ , then eventually  $denom_k$  must hit 0.
- Correctness follows from part (3).
- The induction step is proved by looking at the details of the loop.

18

**Claim:** After two iterations,  $denom$  is halved:

- Recall  $num = q * denom + rem$ . Use  $denom'$  and  $denom''$  to denote value of  $denom$  after 1 and 2 iterations. Two cases:
  1.  $rem \leq denom/2 \Rightarrow denom' \leq denom/2$  and  $denom'' < denom/2$ .
  2.  $rem > denom/2$ . But then  $num' = denom$ ,  $denom' = rem$ . At next iteration,  $q = 1$ , and  $denom'' = rem' = num' - denom' < denom/2$
- How long until  $denom$  is  $\leq 1$ ?
  - $< 2 \log_2(m)$  steps!
- After at most  $2 \log_2(m)$  steps,  $denom = 0$ .

20

## The Extended Euclidean Algorithm

**Theorem 5:** For  $a, b \in N$ , not both 0, we can compute  $s, t \in Z$  such that

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

- **Example:**  $\gcd(9, 4) = 1 = 1 \cdot 9 + (-2) \cdot 4.$

**Proof:** By strong induction on  $\max(a, b)$ . Suppose without loss of generality  $a \leq b$ .

- If  $\max(a, b) = 1$ , then must have  $b = 1$ ,  $\gcd(a, b) = 1$ 
  - $\gcd(a, b) = 0 \cdot a + 1 \cdot b.$
- If  $\max(a, b) > 1$ , there are three cases:
  - $a = 0$ ; then  $\gcd(0, b) = b = 0 \cdot a + 1 \cdot b$
  - $a = b$ ; then  $\gcd(a, b) = a = 1 \cdot a + 0 \cdot b$
  - If  $0 < a < b$ , then  $\gcd(a, b) = \gcd(a, b - a)$ . Moreover,  $\max(a, b) > \max(a, b - a)$ . Thus, by IH, we can compute  $s, t$  such that

$$\gcd(a, b) = \gcd(a, b - a) = sa + t(b - a) = (s - t)a + tb.$$

**Note:** this computation basically follows the “recipe” of Euclid’s algorithm.