Algorithms for Integers

Last Updated: August 30th, 2023

1 Introduction

Integers lie at the very foundation of computing. For example, to run any program on your laptop it must first be translated into one written in *machine language* and consisting of a sequence of binary numbers each of which encodes an instruction. Moreover, each instruction is decoded with the help of arithmetic integer operations, and the instruction itself may call for an arithmetic integer operation to be performend on data that is stored in registers and are represented as a binary integers. In addition, there are fundamental problems involving integers that are central to computer science practice and theory, including the problem of deciding if a number is prime, and the problem of deciding if a Diophantine equation has any solutions.

In this lecture we examine algorithms for basic integer operations, as well as some number-theoretic concepts and properties and algorithms that support them. We finish the lecture with applictions to primality testing, cryptography, and hashing.

2 Algorithms for Arithmetic Operations

In practice most of the arithmetic operations we need to compute are performed in hardware within an arithmetic-logic unit (ALU). However, there are applications, such as cryptography, where the number of bits of each integer far exceeds the 32 or 64-bit limitations of the ALU. In such cases we must perform the operations in software. In what follows we assume that, for each operation, the operands are both n-bit nonnegative binary numbers.

2.1 Addition and Subtraction

For addition and subtraction we may use the elementary-school approach of stacking the numbers to form a $2 \times n$ array. Then starting with the least significant bit (LSB) column, and working from left to right we add the bits of each column, including the carry bit, and place the new carry bit at the top of the next column (moving left to right). This algorithm requires $\Theta(n)$ steps.

Example 2.1. Demonstrate the elementary-school addition algorithm for computing the sum of 55 and 37.

2.2 Multiplication

For multiplication $x \cdot y$ we may also use the elementary school approach that requires making n rows that are successively indented one place to the left. In the case of binary numbers, row i $(0 \leq i < n)$ equals either 0 if $y_i = 0$, or x if $y_i = 1$. But perhaps an easier way (in terms of having to do it by hand), is to use the following recursive approach.

```
Name: multiply
Inputs: two $n$-bit integers $x$ and $y$.
Output: $xy$.
Begin
If $y = 0$, then return 0.
$z = $ multiply$(x, \lfloor y/2 \rfloor)$.
If $y$ is even, then return $2z$.
Else return $x + 2z$.
End
```

It can be shown that both the elementary and recursive approach require a worst-case $O(n^2)$ number of steps (prove this!). However, in a later lecture we'll show there are other multiplication algorithms that perform much better in the worst-case!

Example 2.2. Use the recursive approach to integer multiplication to compute the product xy, where $x = 19$ and $y = 24$.

Solution.

xy	x	y
	19	24
	19	12
	19	6
	19	3
	19	1

2.3 Divison and Mod

We now provide a recursive algorithm for computing the quotient and remainder when dividing one integer by another. Where as our multiplication algorithm recursed on the multiplier y, this algorithm recurses on the dividend x.

```
Name: divide
Inputs: two $n$-bit integers $x$ and $y$.
Output: $(\lfloor x/y \rfloor, x \bmod y)$
Begin
If $x = 0$, then return $(q = 0, r = 0)$.
$(q, r) = \texttt{divide}(\lfloor x/2 \rfloor, y)$.
$q = 2 \cdot q$, $r = 2 \cdot r$.
If $x$ is odd, then $r = r + 1$.
If $r \geq y$, then $r = r - y$, $q = q + 1$.
Return $(q, r)$.
End
```

It can be shown that the above algorithm requires a worst-case $O(n^2)$ number of steps. One important property of the remainder is that it is always less than the divisor y.

Example 2.3. Use the recursive approach to division to compute the quotient and remainder when dividing 283 by 15.

Solution.

q	r	x	y
		283	15
		141	15
		70	15
		35	15
		17	15
		8	15
		4	15
		2	15
		1	15
		0	15

3 Euclid's Algorithm

The greatest common divisor (gcd) of two integers a and b, denoted (a,b), is defined as the largest integer that divides both a and b. For example, $(5,0)=5$, $(26,12)=2$, $(36,24)=12$, and $(27,8)=1$. The gcd of two or more numbers plays a central role in much of number theory. For this reason it seems important to have an efficient algorithm for computing (a,b). Euclid's algorithm is such an algorithm, and makes use of the division algorithm from the previous section. It is a recursive algorithm that relies on the following lemma.

Lemma 3.1. If $b=0$, then $(a,b)=a$, and for $b>0$,

$$(a,b)=(b, a \bmod b).$$

Proof. The case when $b=0$ is immediate by definition. Now consider the case when $b>0$. Then by the division algorithm we have

$$a=bq+r,$$

where $r=a \bmod b<b$. Now let d be a common divisor for both a and b. Then, since d divides both a and b, it also divides any linear combination of a and b (see Exercise 5). In particular, d divides $r=a-bq$. Thus, d divides both b and $r=(a \bmod b)$. Conversely, suppose d divides both b and $r=a \bmod b$. Then d also divides $bq+r=a$. Therefore, a number divides both a and b iff it divides both b and $a \bmod b$, and so $(a,b)=(b, a \bmod b)$. □

```
Name: gcd
Inputs: two $n$-bit integers $a$ and $b$.
Output: $(a,b)$.
Begin
If $b = 0$, then return $a$.
Return gcd$(b, a \bmod b)$.
End
```

The running time of Euclid's algorithm is O(n^3) since computing (q,r) in each round requires $O(n^2)$ steps, and there are at most n rounds since remainder $r=a \bmod b$ has a size that is at least one bit less than the size of b.

Example 3.2. We apply Euclid's algorithm to $a = 3794$ and $b = 2226$.

a	$(b)(q)$	r
3794	(2226)(1)	1568
2226	(1568)(1)	658
1568	(658)(2)	252
658	(252)(2)	154
252	(154)(1)	98
154	(98)(1)	56
98	(56)(1)	42
56	(42)(1)	14
42	(14)(3)	0

Therefore, $(3794, 2226) = 14$. □

One immediate application of Euclid's algorithm is that, by working backwards through the algorithm's equations and making appropriate substitutions, it allows for one to write (a, b) as a linear combination of a and b. Indeed, using the previous example, we have

$$56 - 42(1) = 14 \Longleftrightarrow$$

$$(98 - 42) - 42(1) = 98 - 42(2) = 14 \Longleftrightarrow$$

$$98 - (98 - 56)(2) = 98(-1) + 56(2) = 14 \Longleftrightarrow$$

$$98(-1) + (154 - 98)(2) = 154(2) + 98(-3) = 14 \Longleftrightarrow$$

$$154(2) + (252 - 154)(-3) = 252(-3) + 154(5) = 14 \Longleftrightarrow$$

$$252(-3) + (658 - 252(2))(5) = 658(5) + 252(-13) = 14 \Longleftrightarrow$$

$$658(5) + (1568 - 658(2)(-13) = 1568(-13) + 658(31) = 14 \Longleftrightarrow$$

$$1568(-13) + (2226 - 1568)(31) = 2226(31) + 1568(-44) = 14 \Longleftrightarrow$$

$$2226(31) + (3794 - 2226)(-44) = 3794(-44) + 2226(75) = 14.$$

Therefore,

$$(3794, 2226) = 14 = 3794(-44) + 2226(75).$$

□

Proposition 3.3. $(a, b) \mid c$ iff there exist constants x and y for which

$$ax + by = c.$$

Proof. Suppose $(a, b) \mid c$ is true, i.e., $(a, b)k = c$, for some $k \geq 0$. Then by performing Euclid's backwards algorithm, we know that there exist constants x and y for which

$$ax + by = (a, b),$$

which in turn yields

$$c = (a, b)k = (ax + by)k = (ax)k + (by)k = a(xk) + b(yk)$$

is a linear combination of a and b.

Conversely, if $ax + by = c$ then, since (a, b) divides both a and b, it must also divide the linear combination $ax + by = c$. □

4 Modular Arithmetic

Let a, b, and $m \geq 1$ be integers. Then we say that a is **congruent** to b, **mod** m, written

$$a \equiv b \text{ mod } m,$$

iff $m \mid (a - b)$.

Proposition 4.1. The following statements are equivalent.

1. $a \equiv b \mod m$.
2. $a = mk + b$ for some integer k.
3. $a \text{ mod } m = b \text{ mod } m$

The proof is left as an exercise.

Example 4.2. The following statements are true.

a. $10 \equiv 4 \bmod 3$

b. $31 \equiv -2 \bmod 11$

c. $20 \not\equiv 5 \bmod 10$ □

Proposition 4.3. Congruence mod m is an **equivalence relation**, i.e., the following statements are true.

Reflexive $a \equiv a \bmod m$

Symmetric $a \equiv b \bmod m \Leftrightarrow b \equiv a \bmod m$

Transitive $a \equiv b \bmod m \wedge b \equiv c \bmod m \Rightarrow a \equiv c \bmod m$

Moreover, there are m distinct equivalence classes, and each is called a **residue class**, where $(a)_m$ denotes the residue class containing $a \in \mathcal{I}$.

Proof. We prove transitivity and leave the other two properties as exercises. By definition, $a \equiv b \bmod m$ implies that $m \mid (a-b)$. Similarly, $b \equiv c \bmod m$ implies that $m \mid (b-c)$. Thus,

$$m \mid [(a-b)+(b-c)] \Rightarrow m \mid (a-c) \Rightarrow a \equiv c \bmod m.$$

Finally, by Proposition 4.1, a necessary and sufficient condition for $a \equiv b \bmod m$ is for

$$a \bmod m = b \bmod m,$$

and since there are m possible remainders when dividing by m there must be exactly m residue classes. □

Definition 4.4. A set S of m integers is called a **complete residue system** mod m, iff, for each mod-m residue class $(a)_m$ there is some $b \in S$ for which $b \in (a)_m$.

Proposition 4.5. Suppose we have $x \equiv x' \bmod m$ and $y \equiv y' \bmod m$. Then the following statements are true.

1. $x + y \equiv x' + y' \bmod m$
2. $x - y \equiv x' - y' \bmod m$
3. $xy \equiv x'y' \bmod m$
4. $x^p \equiv (x')^p \bmod m$, for all $p \geq 0$.

Proof. We prove the first property and leave the others as exercises. Since $x \equiv x' \bmod m$ and $y \equiv y' \bmod m$, we have $m \mid (x - x')$ and $m \mid (y - y')$. This in turn implies that

$$m \mid (x - x') + (y - y').$$

Rearranging terms, we have

$$m \mid [(x + y) - (x' + y')].$$

In other words,

$$x + y \equiv x' + y' \bmod m.$$

□

Proposition 4.5 makes it possible to greatly simplify an arithmetic mod-m expression whose operations consist of sums, differences, products, and powers. In general, any number x in the expression may be replaced by any number x' that is congruent to x mod m.

Example 4.6. Show that

$$3^{3n+1}5^{2n+1} + 2^{5n+1}11^n$$

is divisible by 17.

Solution.

Proposition 4.7. Let $d = (c, m)$. Then

$$ac \equiv bc \text{ mod } m \text{ iff } a \equiv b \text{ mod } \frac{m}{d}.$$

Proof. If $ac \equiv bc$ mod m, then

$$m \mid (ac - bc) \Rightarrow m \mid (a-b)c \Rightarrow \frac{m}{d} \mid (a-b)c.$$

Notice that $\frac{m}{d}$ shares no prime factors with c (why?) and so must divide $a - b$ which implies

$$a \equiv b \text{ mod } \frac{m}{d}.$$

Conversely, if

$$a \equiv b \text{ mod } \frac{m}{d},$$

then

$$\frac{m}{d} \mid (a-b)$$

and, since $d \mid c$,

$$(\frac{m}{d})(d) = m \mid (a-b)c$$

which implies

$$m \mid (ac - bc).$$

Therefore,

$$ac \equiv bc \text{ mod } m.$$

□

Proposition 4.8. The equation

$$ax \equiv b \bmod m$$

has a solution iff $(a, m) \mid b$.

The proof is left as an exercise.

We say that a has a multiplicative inverse b, mod m, iff $ab \equiv 1 \bmod m$.

Corollary 4.9. Integer a has a multiplicative inverse mod m iff $(a, m) = 1$, i.e. iff a is **relatively prime** to m.

Proof. Integer a having a multiplicative inverse is equivalent to the equation $ax \equiv 1 \bmod m$ having a solution. But By Proposition 4.8, this equation has a solution iff $(a, m) \mid 1$, which can only occur when $(a, m) = 1$. □

Example 4.10. Compute the multiplicative inverse of 5 mod 17.

Proposition 4.11. If $p \geq 2$ is prime and $a \not\equiv 0 \bmod p$, then

$$a \cdot 1 \bmod p, a \cdot 2 \bmod p, \ldots, a \cdot (p-1) \bmod p$$

is a permutation of the numbers $1, 2, \ldots, p-1$.

Proof. By Corollary 4.9, a has a multiplicative inverse b mod p. Then for $i \in \{1, \ldots, p-1\}$ we have that $a \cdot i \equiv 0 \bmod p$ implies that

$$b(ai) \equiv (ba)i \equiv (ab)i \equiv 1 \cdot i \equiv i \equiv b \cdot 0 \equiv 0 \bmod p,$$

which implies $i \equiv 0 \bmod p$, a contradiction. Also, if $a \cdot i \equiv a \cdot j \bmod p$, then

$$i \equiv 1 \cdot i \equiv (ab)i \equiv (ba)i \equiv b(ai) \equiv b(aj) \equiv (ba)j \equiv (ab)j \equiv 1 \cdot j \equiv j \bmod p.$$

Thus $a \cdot i \bmod p$ and $a \cdot j \bmod p$ must be two distinct numbers so long as $i \neq j$. Hence,

$$a \cdot 1 \bmod p, a \cdot 2 \bmod p, \ldots, a \cdot (m-1) \bmod p$$

is a sequence of $p-1$ distinct numbers from the set $\{1, \ldots, p-1\}$. In other words, the sequence is a permutation of $1, \ldots, p-1$. $\square$

Theorem 4.12. (Fermat's Little Theorem) If p is prime and $1 \leq a < p$, then $a^{p-1} \equiv 1 \bmod p$.

Proof. Since p is prime, by Proposition 4.11, we know that

$$(a \cdot 1)(a \cdot 2) \cdots (a \cdot (p-1)) \equiv a^{p-1}(p-1)! \equiv (p-1)! \bmod p,$$

and so

$$p \mid (a^{p-1}(p-1)! - (p-1)!.$$

But this is equivalent to

$$p \mid (a^{p-1} - 1)(p-1)!$$

and, since $p \not| \, (p-1)!$, we have $p \mid (a^{p-1} - 1)$, i.e., $a^{p-1} \equiv 1 \bmod p$. □

5 Jacobi Symbols

Let a and m be integers with $(a, m) = 1$ and $m \geq 3$ odd. Then the **Jacobi symbol** $\left(\frac{a}{m}\right)$ takes on the value of either 1 or -1 and has several applications in number theory. Perhaps its most famous use occurs when m is prime and is used to determine if a is a **quadratic residue** mod m, i.e. if there is a number x for which $x^2 \equiv a$ mod m. It turns out that a is a mod-m quadratic residue iff $\left(\frac{a}{m}\right) = 1$.

The next section provides another application of the Jacobi symobl, as it is used in the Strassen-Solovay randomized algorithm for testing if a number is prime.

The following theorem provides the rules to follow when computing $\left(\frac{a}{m}\right)$. It is stated without proof.

Theorem 5.1. Each of the following formulas is valid so long as the "denominator" is an odd integer greater than 1, and is co-prime with the "numerator". In case of (v), we also assume that the "numerator" n is also an odd number greater than 1.

(i) $a \equiv b \pmod{m} \Rightarrow \left(\frac{a}{m}\right) = \left(\frac{b}{m}\right)$

(ii) $\left(\frac{ab}{m}\right) = \left(\frac{a}{m}\right)\left(\frac{b}{m}\right)$ $\qquad$ $\left(\frac{a}{mn}\right) = \left(\frac{a}{m}\right)\left(\frac{a}{n}\right)$

(iii)
$$\left(\frac{-1}{m}\right) = \begin{cases} 1 & \text{if } m \equiv 1 \pmod{4} \\ -1 & \text{if } m \equiv -1 \pmod{4} \end{cases}$$

(iv)
$$\left(\frac{2}{m}\right) = \begin{cases} 1 & \text{if } m \equiv \pm 1 \pmod{8} \\ -1 & \text{if } m \equiv \pm 3 \pmod{8} \end{cases}$$

(v)
$$\left(\frac{m}{n}\right) = \begin{cases} -\left(\frac{n}{m}\right) & \text{if } m \equiv n \equiv -1 \pmod{4} \\ \left(\frac{n}{m}\right) & \text{otherwise.} \end{cases}$$

Example 5.2. Compute $\left(\frac{2342}{11239}\right)$. Hint: 11239 is a prime number.

6 A Randomized Algorithm for Primality Testing

A **randomized algorithm** is one that allows for function calls of the form `random`(a, b), where $a \leq b$ are integers (or possibly real numbers), and for which an oracle randomly selects an integer x, $a \leq x \leq b$ as the returned value. This selection makes use of the uniform distribution and is independent of any previous selections made by the oracle. Randomized algorithms are often characterized by not always being guaranteed to return the correct answer, but have a probability (usually greater than 0.5) of returning a correct answer. Of course, in practice, oracles do not exist, and so the `random` fuunction must be implemented using a **pseudorandom number generator**, i.e., a deterministic algorithm that generates a sequence of integers which appears statistically similar to a theoretically random sequence.

In this section we describe a randomized algorithm, called the Solovay-Strassen Test, for deciding if some odd positive integer is a prime number. The algorithm is based on the following theorem which is stated without proof.

Theorem 6.1. Let $n > 1$ be an odd integer and consider the congruence

$$a^{\frac{n-1}{2}} \equiv \left(\frac{a}{n}\right) \bmod n, \tag{1}$$

where $\left(\frac{a}{n}\right)$ denotes the Jacobi symbol. If n is prime, then *every* $a \not\equiv 0 \bmod n$ satisfies the congruence. However, if n is composite, then less than half of the members of a complete residue system will satisfy the congruence. An a value that does not satisfy the congruence is called a **witnesses** because finding just one is evidence that n is composite. Otherwise, a is called an **accomplice**.

Theorem 6.1 yields the following randomized algorithm for deciding if a positive integer $n > 2$ is prime.

```
Name: is_prime

Input: integer $n > 2$

Output: 1 iff $n$ is estimated to be prime.

Begin

If $n$ is even, then return 0.

Do the following 1000 times:

    $a = \text{random}(2, n-1)$. //Randomly select an integer from $[2, n-1]$.
    If $(a, n) > 1$, then return 0.
    If $a^{\frac{n-1}{2}} \not\equiv \left(\frac{a}{n}\right) \mod n$, then return 0. //$a$ is a witness

Return 1.

End
```

Theorem 6.2. The above algorithm returns the correct output with probability at least $1 - \left(\frac{1}{2}\right)^{1000}$.

Proof. The algorithm is certainly correct by returning 0 when n is even, and when returning 0 when an $a \in [2, n-1]$ is discovered for which $(a, n) > 1$ (why ?). Also, by Theorem 6.1, the algorithm also correctly returns 0 when a does not satisfy Equation 1. Hence, the only way for an error to occur is at the final line when 1 is returned. Furthermore, the only way an error may occur is if n is composite. But if n is composite then, by Theorem 6.1, in each iteration of the `do`-loop, with at least probability 0.5 a witness a will be selected, in which case 0 is returned. Moreover, since we may assume that all 1000 a-values are independently generated, this gives a probability of at most $\left(\frac{1}{2}\right)^{1000}$ that each of the a values will be an acccomplice, which erroneosly causes 1 to be returned. Therefore, with probability at least $1 - \left(\frac{1}{2}\right)^{1000}$, the correct answer of 1 is returned. □

Example 6.3. Verify that for $n = 9$, there are at least $\lceil (9-2)/2 \rceil = 4$ witnesses a between 2 and 8 to the fact that $a^4 \not\equiv \left(\frac{a}{9}\right) \bmod 9$.

7 RSA Scheme of Public-Key Cryptography

In public key cryptography, if Alice wants to send a secure message to Bob, then Bob provides Alice a **public key** that allows her to encrypt a message. Anyone with the public key is able to encrypt a message, but only Bob has the **private key** that is needed to decrypt any encrypted message. Furthermore, in the **RSA** scheme we may think of Alice's message as a number x, and Bob's public key is the pair (N, e), where $N = pq$ is the product of two (very large!) distinct primes p and q, and e is relatively prime to $(p-1)(q-1)$. Then the encrypted message is also a number, namely $x^e \bmod N$.

Theorem 7.1. The following two statements are true about $x^e \bmod N$.

1. $x_1^e \equiv x_2^e \bmod N$ implies $x_1 \equiv x_2 \bmod N$. In other words, $x^e \bmod N$ is a bijection (both one-to-one and onto).
2. e has a multiplicative inverse $d \bmod (p-1)(q-1)$, so that
$$(x^e)^d \equiv x \bmod N,$$
meaning that d may be used to decrypt the message.

Proof. Suppose Statement 2. is true and $x^e \bmod N$ is not a bijection. Then there must be $x_1, x_2, y \in [0, N-1]$, $x_1 \neq x_2$, for which $x_1^e \equiv y \equiv x_2^e \bmod N$. But then

$$x_1 \equiv (x_1^e)^d \equiv y^d \equiv (x_2^e)^d \equiv x_2 \bmod N,$$

which is a contradiction. Thus it suffices to prove Statement 2.

Since e is relatively prime with $(p-1)(q-1)$, e has a multiplicative inverse, call it d, modulo $(p-1)(q-1)$ and so there exists integer k for which

$$ed = 1 + k(p-1)(q-1).$$

Moreover, since our goal is to show that $x^{ed} \equiv x \bmod N$, by the definition of congruence mod N it suffices to show that N divides

$$x^{ed} - x = x^{1+k(p-1)(q-1)} - x.$$

By Fermat's Little Theorem,

$$x^{ed} - x \equiv x^{1+k(p-1)(q-1)} - x \equiv x \cdot (x^{k(q-1)})^{(p-1)} - x \equiv x \cdot 1 - x \equiv 0 \bmod p.$$

By similar reasoning, we also have

$$x^{ed} - x \equiv 0 \bmod q.$$

Thus, both p and q divide $x^{ed} - x$ and therefore $N = pq$ divides it as well, since $x^{ed} - x$ has both a p factor and a q factor. □

Example 7.2. If $N = (5)(7) = 35$ and $e = 11$, determine the the decryption key d. Verfiy that d correctly decrypts the encryption of the message $x = 22$.

Exercises

1. Demonstrate the recursive multiplication algorithm using $x = 42$ and $y = 25$.

2. Demonstrate the recursive multiplication algorithm using $x = 59$ and $y = 36$.

3. Demonstrate the recursive division algorithm using $x = 144$ and $y = 12$.

4. Demonstrate the recursive division algorithm using $x = 278$ and $y = 29$.

5. Prove that if $x \mid y$ and $x \mid z$, then $x \mid (ay + bz)$, for any integers a and b.

6. Demonstrate Euclid's algorithm using $a = 195$ and $b = 130$. Then work backwards through the algorithm to write $(195, 130)$ as a linear combination of 195 and 130.

7. Demonstrate Euclid's algorithm using $a = 2037$ and $b = 1533$. Then work backwards through the algorithm to write $(2037, 1533)$ as a linear combination of 2037 and 1533.

8. Prove Proposition 4.1.

9. Which of the following statements are true.

   a. $-3 \equiv 7 \bmod 5$
   b. $-11 \equiv -7 \bmod 9$
   c. $17 \equiv -17 \bmod 11$
   d. $0 \equiv 14 \bmod 7$

10. Provide ten members each of $(13)_7$ and $(-5)_{11}$.

11. Prove the congruence mod m is both a reflexive and symmetric relation on the set $\mathcal{I}$ of integers.

12. Prove that
$$61^{k+1} + 11^k 7^{2k} 3^{3k} 2^{5k+3}$$
is divisible by 23.

13. Prove that
$$4^{1536} - 9^{4824}$$
is divisible by 35.

14. Simplify
$$2^{2^{2006}} \bmod 3.$$

15. Use Propostion 3.3 to prove Proposition 4.8.

16. Determine the multiplicative inverse of 37 mod 43.

17. Determine the multiplicative inverse of 16 mod 25.

18. Compute the following Jacobi symbols.

   a. $\left(\frac{1234567}{225}\right)$

b. $\left(\frac{31}{95}\right)$

c. $\left(\frac{589}{1999}\right)$

d. $\left(\frac{1113}{11131}\right)$

19. Verify that for $n = 7$ that, for all $a \in [2, 6]$, $a^3 \equiv \left(\frac{a}{7}\right) \mod 7$.

20. Verify that for $n = 15$, there are at least $\lceil (15 - 2)/2 \rceil = 7$ witnesses a between 2 and 14 to the fact that $a^7 \not\equiv \left(\frac{a}{15}\right) \mod 15$.

21. For an RSA cryptosystem, $p = 7$ and $q = 11$. Determine appropriate e and d.

22. Consider an RSA key set with $p = 17$, $q = 23$, and $N = 391$, and $e = 3$. Determine the value of d that should be used for the decryption key. Determine the encryption of the message $M = 41$. Verify that it is correctly decrypted using the d you calculated.

Exercise Solutions

1. Demonstrate the recursive multiplication algorithm using $x = 42$ and $y = 25$.

2. Demonstrate the recursive multiplication algorithm using $x = 59$ and $y = 36$.

3. Demonstrate the recursive division algorithm using $x = 144$ and $y = 12$.

4. Demonstrate the recursive division algorithm using $x = 278$ and $y = 29$.

5. Prove that if $x \mid y$ and $x \mid z$, then $x \mid (ay + bz)$, for any integers a and b.

6. Demonstrate Euclid's algorithm using $a = 195$ and $b = 130$. Then work backwards through the algorithm to write $(195, 130)$ as a linear combination of 195 and 130.

7. Demonstrate Euclid's algorithm using $a = 2037$ and $b = 1533$. Then work backwards through the algorithm to write $(2037, 1533)$ as a linear combination of 2037 and 1533.

8. $1 \to 2$. If $a \equiv b \bmod m$, then there is a k for which $mk = a - b$, in which case $a = mk + b$.

   $2 \to 3$. Suppose $a = mk + b$. Then by the division algorithm we may write $b = mq + r$, where $0 \le r < m$. But then we have
   $$a = mk + mq + r = m(k + q) + r,$$
   and so both a and b have the same remainder when divided by m, i.e., $a \bmod m = b \bmod m$.

   $3 \to 1$. Since $a \bmod m = b \bmod m$, we have $a = mk_1 + r$ and $b = mk_2 + r$. Thus,
   $$(a - b) = mk_1 + r - mk_2 - r = mk_1 - mk_2 = m(k_1 - k_2),$$
   which, by definition, implies $a \equiv b \bmod m$.

9. a and d are true.

10. $6, 13, 20, 27, 34, -1, -8, -15, -22, -29 \in (13)_7$ and
    $$6, 17, 28, 39, 50, -5, -16, -27, -38, -49 \in (-5)_{11}.$$

11. $a \equiv a \bmod m$ since $m \mid (a-a)$, and if $a \equiv b \bmod m$, then $m \mid (a-b)$, in which case $m \mid (b-a)$, so that $b \equiv a \bmod m$.

12. Prove that
    $$61^{k+1} + 11^k 7^{2k} 3^{3k} 2^{5k+3}$$
    is divisible by 23.

13. Prove that
    $$4^{1536} - 9^{4824}$$
    is divisible by 35.

14. Simplify
    $$2^{2^{2006}} \bmod 3.$$

15. Use Propostion 3.3 to prove Proposition 4.8.

16. Determine the multiplicative inverse of 37 mod 43.

17. Determine the multiplicative inverse of 16 mod 25.

18. Compute the following Jacobi symbols.

    a. $\left(\frac{1234567}{225}\right)$
    b. $\left(\frac{31}{95}\right)$
    c. $\left(\frac{589}{1999}\right)$
    d. $\left(\frac{1113}{11131}\right)$

19. Verify that for $n = 7$ that, for all $a \in [2, 6]$, $a^3 \equiv \left(\frac{a}{7}\right) \mod 7$.

20. Verify that for $n = 15$, there are at least $\lceil (15-2)/2 \rceil = 7$ witnesses a between 2 and 14 to the fact that $a^7 \not\equiv \left(\frac{a}{15}\right) \mod 15$.

21. For an RSA cryptosystem, $p = 7$ and $q = 11$. Determine appropriate e and d.

22. Consider an RSA key set with $p = 17$, $q = 23$, and $N = 391$, and $e = 3$. Determine the value of d that should be used for the decryption key. Determine the encryption of the message $M = 41$. Verify that it is correctly decrypted using the d you calculated.