Exercises

1. Trace Algorithm 1 when it is given \( m = 5 \) as input. That is, show all steps used by Algorithm 1 to find 51, as is done in Example 1 to find 41.

2. Trace Algorithm 1 when it is given \( m = 6 \) as input. That is, show all steps used by Algorithm 1 to find 61, as is done in Example 1 to find 41.

3. Trace Algorithm 3 when it finds gcd(8, 13). That is, show all the steps used by Algorithm 3 to find gcd(8, 13).

4. Trace Algorithm 3 when it finds gcd(12, 17). That is, show all the steps used by Algorithm 3 to find gcd(12, 17).

5. Trace Algorithm 4 when it is given \( m = 5, n = 11 \), and \( b = 3 \) as input. That is, show all the steps Algorithm 4 uses to find \( 31 \mod 5 \).

6. Trace Algorithm 4 when it is given \( m = 7, n = 10 \), and \( b = 2 \) as input. That is, show all the steps Algorithm 4 uses to find \( 21 \mod 7 \).

7. Give a recursive algorithm for computing \( mn \) whenever \( n \) is a positive integer and \( x \) is an integer, using just addition.

8. Give a recursive algorithm for finding the sum of the first \( n \) odd positive integers.

9. Give a recursive algorithm for finding the sum of the first \( n \) positive integers.

10. Give a recursive algorithm for finding the maximum of a finite set of integers, making use of the fact that the maximum of \( n \) integers is the larger of the last integer in the list and the maximum of the first \( n - 1 \) integers in the list.

11. Give a recursive algorithm for finding the minimum of a finite set of integers, making use of the fact that the minimum of \( n \) integers is the smaller of the last integer in the list and the minimum of the first \( n - 1 \) integers in the list.

12. Devise a recursive algorithm for finding \( x \mod m \) whenever \( x \) and \( m \) are positive integers based on the facts that \( x \mod m = (x - m \cdot \lfloor x \div m \rfloor) \mod m \).

13. Give a recursive algorithm for finding \( x \mod m \) whenever \( x \) and \( m \) are positive integers.

14. Give a recursive algorithm for finding a \( \text{mode} \) of a list of integers. (A \( \text{mode} \) is an element in the list that occurs at least as often as any other element.)

15. Devise a recursive algorithm for computing the greatest common divisor of two nonnegative integers \( a \) and \( n \) with \( a \geq b \) using the fact that \( \gcd(a, b) = \gcd(b, a - b) \).

16. Prove that the recursive algorithm for finding the sum of the first \( n \) positive integers you found in Exercise 8 is correct.
17. Describe a recursive algorithm for multiplying two non-negative integers \(a\) and \(b\) based on the fact that \(x \times y = 2(x \times (y/2)) + x\) when \(y\) is even and \(x \times y = 2(x \times ((y-1)/2)) + x\) when \(y\) is odd, together with the initial condition \(x \times y = 0\) when \(y = 0\).
18. Prove that Algorithm 1 for computing \(a^n\) when \(a\) is a non-negative integer is correct.
19. Prove that Algorithm 5 for computing \((a+b, h)\) when \(a\) and \(b\) are positive integers with \(a < b\) is correct.
20. Prove that the algorithm you devised in Exercise 17 is correct.
21. Prove that the recursive algorithm that you found in Exercise 17 is correct.
22. Prove that the recursive algorithm that you found in Exercise 10 is correct.
23. Devise a recursive algorithm for computing \(a^n\) where \(n\) is a non-negative integer, using the fact that \((a+1)^n = a^n + 2n + 1\). Then prove that this algorithm is correct.
24. Devise a recursive algorithm to find \(a^2\), where \(a\) is a real number and \(n\) is a positive integer. \(\text{Hint: Use the equality } a^{2^n} = (a^2)^{2^{n-1}}.\)
25. How does the number of multiplications used by the algorithm in Exercise 24 compare to the number of multiplications used by Algorithm 2 to evaluate \(a^n\) ?
26. Use the algorithm in Exercise 24 to devise an algorithm for evaluating \(a^n\) when \(n\) is a non-negative integer. \(\text{Hint: Use the binary expansion of } n.\)
27. How does the number of multiplications used by the algorithm in Exercise 26 compare to the number of multiplications used by Algorithm 2 to evaluate \(a^n\)?
28. How many additions are used by the recursive and iterative algorithms given in Algorithms 7 and 8, respectively, to find the Fibonacci number \(n\)?
29. Devise a recursive algorithm to find the \(x\)th term of the sequence defined by \(a_0 = 1, a_1 = 2, a_{n+2} = 2a_{n+1} + a_n\), for \(n = 2, 3, 4, \ldots\).
30. Devise an iterative algorithm to find the \(x\)th term of the sequence defined in Exercise 29.
31. Is the recursive or the iterative algorithm for finding the sequence in Exercise 29 more efficient?
32. Devise a recursive algorithm to find the \(x\)th term of the sequence defined by \(a_0 = 1, a_1 = 2, a_2 = 3, a_{n+2} = a_{n+1} + a_n\), for \(n = 3, 4, 5, \ldots\).
33. Devise an iterative algorithm to find the \(x\)th term of the sequence defined in Exercise 32.
34. Is the recursive or the iterative algorithm for finding the sequence in Exercise 32 more efficient?
35. Give iterative and recursive algorithms for finding the \(x\)th term of the sequence defined by \(a_0 = 1, a_1 = 3, a_2 = 5, a_{n+2} = a_{n+1} + a_n\), for \(n = 3, 4, 5, \ldots\). Which is more efficient?
36. Give a recursive algorithm to find the number of partitions of a positive integer based on the recursive definition given in Exercise 47 in Section 5.3.
37. Give a recursive algorithm for finding the reversal of a bit string. (See the definition of the reversal of a bit string in the preamble of Exercise 34 in Section 5.3.)