§ 2.2. Complexity of Algorithms

7. The conventional algorithm for evaluating a polynomial \( a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 \) at \( x = c \) can be expressed in pseudocode by

```
procedure polynomial (c, a_0, a_1, ..., a_n: real numbers)
    power := 1
    y := a_0
    for i := 1 to n
        begin
            power := power * c
            y := y + a_i * power
        end
    {y = a_n c^n + a_{n-1} c^{n-1} + \cdots + a_1 c + a_0}
```

where the final value of \( y \) is the value of the polynomial at \( x = c \).

(a) Evaluate \( 3x^2 + x + 1 \) at \( x = 2 \) by working through each step of the algorithm.

(b) Exactly how many multiplications and additions are used to evaluate a polynomial of degree \( n \) at \( x = c \)? (Do not count additions used to increment the loop variable.)

Solution.
(a) The for-loop is executed twice. Before entering the for-loop, \( \text{power} = 1 \) and \( y = 1 \); then after the first loop \( (i = 1) \), \( \text{power} = 2 \), \( y = 1 + 1 \times 2 = 3 \); after the second loop \( (i = 2) \), \( \text{power} = 4 \), \( y = 3 + 3 \times 4 = 15 \), which is returned as the final value of the polynomial \( ax^2 + x + 1 \) at \( x = 2 \).

(b) Every time the for-loop is executed, two multiplications and one addition are operated, and the for-loop is executed \( n \) times while the code is used to evaluate a polynomial of degree \( n \). Therefore the answer is \( 2n \) multiplications and \( n \) additions are used.

8. There is a more efficient algorithm (in terms of the number of multiplications and additions used) for evaluating polynomials than the conventional algorithm described in the previous exercise. It is called Horner’s method. The following pseudocode shows how to use this method to find the value of \( a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 \) at \( x = c \).

```plaintext
procedure Horner (c, a_0, a_1, a_2, \ldots, a_n: real numbers)
    y := a_n
    for i := 1 to n
        y := y * c + a_{n-i}
    {y = a_n c^n + a_{n-1} c^{n-1} + \cdots + a_1 c + a_0}
```

(a) Evaluate \( 3x^2 + x + 1 \) at \( x = 2 \) by working through each step of the algorithm.

(b) Exactly how many multiplications and additions are used to evaluate a polynomial of degree \( n \) at \( x = c \)? (Do not count additions used to increment the loop variable.)

Solution.

(a) The for-loop is executed twice. Before entering the for-loop, \( y = 3 \); then after the first loop \( (i = 1) \), \( y = 3 \times 2 + 1 = 7 \); after the second loop \( (i = 2) \), \( y = 7 \times 2 + 1 = 15 \), which is returned as the final value of the polynomial \( ax^2 + x + 1 \) at \( x = 2 \).

(b) Every time the for-loop is executed, only one multiplications and one addition are operated, and the for-loop is executed \( n \) times while the code is used to evaluate a polynomial of degree \( n \). Therefore the answer is \( n \) multiplications and \( n \) additions are used.

10. How much time does an algorithm take to solve a problem of size \( n \) if this algorithm uses \( 2n^2 + 2^n \) bit operations, each requiring \( 10^{-9} \) second, with the following values for \( f(n) \)?

(a) 10  (b) 20  (c) 50  (d) 100

Solution.

(a) \( 0.12240 \times 10^{-5} \) s
(b) \( 0.1049376 \times 10^{-2} \) s
(c) \( 1125899.906847624 \approx 0.11259 \times 10^7 \) s \( \approx 13.031 \) (day)
(d) \( 1267650600228229401496.703225376 \approx 0.12677 \times 10^{22} \) s \( \approx 0.40197 \times 10^{14} \) (year)

13. Analyze the average-case performance of the linear search algorithm, if exactly half the time element \( x \) is not in the list and if \( x \) is in the list it is equally likely to be in any position.
Solution. In general, if \( x \) is the \( i \)-th term of the list, \( 2i + 1 \) comparisons are needed (see EXAMPLE 4 in section 2.2.). If \( x \) is not in the list, it takes \( 2n + 2 \) steps to see it. So suppose the linear search algorithm is executed \( 2n \) times, in which half the time element \( x \) is not in the list and half is in and if \( x \) is in the list it is equally likely to be in any position, then average number of comparisons used is
\[
\frac{3 + 5 + 7 + \cdots + (2n + 1)}{2n} + n \times (2n + 2) = \frac{n + 2 + (2n + 2)}{2} = \frac{3n + 4}{2}.
\]

§ 2.3 The Integers and Division

8. Are the following integers primes?

(a) 19  
(b) 27  
(c) 93  
(d) 101  
(e) 107  
(f) 113

Solution.

(a) Yes.  
(b) No, since 27 = 3\(^3\).  
(c) No, since 93 = 3 \times 31.  
(d) Yes.  
(e) Yes.  
(f) Yes.

16. We call a positive integer **perfect** if it equals the sum of its positive divisors other than itself.

(a) Show that 6 and 28 are perfect.

(b) Show that \( 2^{p-1}(2^p - 1) \) is a perfect number when \( 2^p - 1 \) is prime.

Solution.

(a) The divisors of 6 other than itself are 1, 2, 3, and the divisors of 28 other than itself are 1, 2, 4, 7, 14. Easy to verify that both 6 and 28 are perfect numbers.

(b) Assume \( 2^p - 1 \) is prime. Then the divisors of \( 2^{p-1}(2^p - 1) \) other than itself are
\[
1, 2, 2^2, \ldots, 2^{p-2}, 2^{p-1}, 1 \cdot (2^p - 1), 2 \cdot (2^p - 1), 2^2 \cdot (2^p - 1), \ldots, 2^{p-2} \cdot (2^p - 1).
\]

The sum = \((1 + 2 + \cdots + 2^{p-2}) \cdot (1 + 2^p - 1) + 2^{p-1} = (2^{p-1} - 1) \cdot 2^p + 2^{p-1} = 2^{p-1}(2^p - 1)\). So \( 2^{p-1}(2^p - 1) \) is a perfect number.

24. What are the greatest common divisors of the following pairs of integers?
(a) $2^2 \cdot 3^3 \cdot 5^2, \quad 2^5 \cdot 3^3 \cdot 5^2$
(b) $2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13, \quad 2^{11} \cdot 3^9 \cdot 11 \cdot 17^{14}$
(c) 17, 17

Solution.

(a) $2^2 \cdot 3^3 \cdot 5^2$
(b) 2 · 3 · 11
(c) 17

(d) $2^2 \cdot 7, \quad 5^3 \cdot 13$
(e) 0, 5
(f) $2 \cdot 3 \cdot 5 \cdot 7, \quad 2 \cdot 3 \cdot 5 \cdot 7$

34. Show that if $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, where $a$, $b$, $c$, $d$, and $m$ are integers with $m \geq 2$, then $a - c \equiv b - d \pmod{m}$.

Solution. Assume $a = b + k \cdot m$ and $c = d + l \cdot m$ for some integers $k, l$. Then $a - c = (b + km) - (d + lm) = (b - d) + (k - l) \cdot m$, so $a - c \equiv b - d \pmod{m}$.

42. What sequence of pseudorandom numbers is generated using the linear congruential generator $x_{n+1} = (4x_n + 1) \mod{7}$ with seed $x_0 = 3$?

Solution. 3, 6, 4, 3, 6, 4, 3, . . .