(vi) *Returning results* is shown as:

```
Return(Value(s))
```

(vii) *While construct* is used as:

```
While condition Do
...
...
Endwhile
```

(viii) *Case construct* is used as:

```
Case
...
Endcase
```

(ix) *Comment* is shown as:

```
/*   Text    */
```

(x) *Parallel activities* by different computing elements are shown through the following constructs in the pseudocode for parallel algorithms:

```
For each processor i, Do in parallel
...
...
Enddo
For all k Do in parallel
...
...
Enddo
For i = 0 to n - 1 and j = 0 to n - 1, Do in parallel
...
...
Enddo
```

SUMMARY

In this chapter, we introduced the general notion of algorithms and the means to characterize asymptotic complexities of algorithms. Linear recurrence relations are quite common in complexity analysis of algorithms. We have indicated how these recurrences are solved using the standard techniques. The basic data structures, used in the algorithm design methods were also briefly described. This chapter also described the conventions used in the representation of algorithms through pseudocode.

EXERCISES

1.1 Develop an algorithm to construct the first N rows of Pascal's triangle. The first four rows of Pascal's triangle are given by:

$$\begin{array}{ccccccc} & & & 1 & & & \\ & & 1 & & 1 & & \\ & 1 & & 2 & & 1 & \\ 1 & & 3 & & 3 & & 1 \end{array}$$

Analyze the algorithm for time complexity.

1.2 Which of the following statements are true? Prove your answers.

$$n^2 \in O(n^3),\ n^2 \in \Omega(n^3),\ 2^n \in \Theta(2^{n+1}),\ n! \in \Theta((n+1)!)$$

1.3 If the Fibonacci sequence, defined as: $f(i) = f(i-1) + f(i-2)$, $(n > 1)$ and $f(0) = f(1) = 1$ is computed by recursion, how many additions will be performed when computing $f(n)$?

1.4 Solve the following recurrence:

$T(N) = T(N-1) + 2T(n-2) - 2T(N-3)$, $N \geq 3$ and

$T(N) = 9N^2 - 15N + 106$, $N = 0, 1, 2$.

1.5 Solve the following recurrence assuming N as an integer power of 2:

$T(N) = 4T(N/2) + N$, $N \geq 2$ and

$T(N) = 1$, $N = 1$.

1.6 Ackermann's function is defined as

$$\begin{aligned} \text{Ack}(i, j) &= j + 1,\ i = 0 \\ &= \text{Ack}(i-1, 1),\ i > 0,\ j = 0 \\ &= \text{Ack}(i-1, \text{Ack}(i, j-1)), \text{ otherwise.} \end{aligned}$$

Calculate Ack(2,3), Ack(3,3).

1.7 Given $K = K_1 + K_2$, find the minimum value of $K_1 \log_2 K_1 + K_2 \log_2 K_2$.

1.8 Consider the following recursive procedure:

```
Procedure Comb(N,K)
  If K = 0 or K = N Then Return(1);
    Else Return(Comb(N - 1, K - 1) + Comb(N - 1, K));
  Endif
End Comb
```

Show that the total number of recursive invocations of Comb( ) is $2\,{}^NC_K - 2$ for computation of Comb(N, K).

1.9 Show that: $f_1(x) = 7.2x^2 + 78x + 10^6$ is $O(x^2)$, $f_2(x) = 2^x/(10^5) - 10$ is $O(2^x)$, and $f_3(x) = x!$ is $O(x^x)$.

1.10 Algorithm A requires n^2 days and algorithm B requires n^3 seconds to solve a problem. Which algorithm would you prefer for a problem instance with $n = 10^6$?

1.11 Sorting method A runs in $8n^2$ steps while sorting method B runs in $64n \log n$ steps. For which values of n does method A run faster than method B?

1.12 Consider the following algorithm:

```
Procedure Test(N)
  For i = 1 to N Do
    For j = 1 to i
      Write i, j, N;
    Endfor
  Endfor
  If N > 0 Then Do
    For i = 1 to 8 Do
      Call Test (N/2);
    Endfor
  Endif
End Test
```

Let $T(N)$ denote the number of lines written by Test(N). Formulate a recurrence relation for $T(N)$.

1.13 Show that $\log^3 n$ is $o(n^{1/3})$.

1.14 What is the time complexity of the following algorithm in terms of 'Big Oh' notation?

```
Procedure Test(N)
  S = 0;
  For I = 1 to N³ Do
    For J = 1 to I Do
      S = S + I;
    Endfor
  Endfor
End Test
```

1.15 Show that $(n + 1)^5$ is $O(n^5)$, 2^{n+1} is $O(2^n)$, $n^3 \log n$ is $\Omega(n^3)$.

1.16 If the space is at a premium, would you use the adjacency list or the adjacency matrix representation to store (i) a graph with 10^4 vertices and 10^4 edges, (ii) a graph with 10^4 vertices and 10^7 edges?

1.17 Solve the recurrence relation, where N is an integer power of 3:

$$T(N) = 6T(N/3) + 2N - 1, N > 1 \text{ and } T(N) = 2, N = 1.$$

1.18 Write an algorithm to check whether every sequence of consecutive ones in a 1-dimensional Boolean array is even. Determine the time complexity as well.

1.19 Write an algorithm to generate all permutations of 1, 2, 3, ..., n. What is the time complexity of the algorithm?

1.20 Arrange the following functions from the lowest asymptotic order to the highest asymptotic order:

$7n$, 2^n, $10n \log n$, $4n^3$, $5n^2$, $2 \log n$, $10n - n^3 + 9n^5$, $n^2 + 7 \log n$.

1.21 Solve the following recurrence where N is an integer power of 2:

$T(N) = 2T(N/2) + \log N,\ N > 1$ and $T(N) = 1,\ N = 1$.

1.22 Develop and analyze an algorithm to determine whether a given $N \times N$ matrix A has the metric property (that is, for all values of $1 \le i, j, k \le N$, $a_{ij} \le a_{ik} + a_{kj}$) or not.

1.23 Solve the recurrence relation: $T(N) = 4T(N/2) + N^3$.

1.24 For two sequences a_n, b_n consider the statements: $a_n = O(b_n)$, $a_n = o(b_n)$. Which one implies the other?

1.25 Find the complexity of the following algorithm:

```
Procedure Test2(n)
  Integer n;
  If n ≤ 1 Then Return(n);
    Else Return (Test2(n - 1) + Test2(n - 2));
  Endif
End Test2
```

1.26 Find the complexity of the following algorithm:

```
Procedure Test3(m,n)
  Integer m,n;
  If n = 0 Then Return(m);
    Else Return(Test3(n, m mod n));
  Endif
End Test3
```

1.27 Given a polynomial, $P = (a_n, a_{n-1}, \ldots, a_0)$, write an algorithm to find $P(x)$ for $x = 2^i$, $0 \le i \le n - 1$. Find the time complexity as well.

1.28 Find the solution of the following recurrence relation in O-notation

$$T(n) = 8T(n/2) + 3n^2$$

where n is an integer power of 2 and greater than 1.

1.29 Show that the solution to the following recurrence is bounded from the lower side by $n \log n$. A tree structure may be used for the proof.

$$T(n) = T(n/3) + T(2n/3) + n$$

1.30 Solve the recurrence: $T(n) = T(n - 1) + T(n - 3) - T(n - 4)$, $n \ge 4$ subject to $T(n) = n$ for $0 \le n \le 3$.

1.31 Solve the recurrence: $T(n) = 3T^2(n - 1)$, for $n \ge 1$ and $T(0) = 1$ using change of variable.

1.32 Show the steps in the construction of a heap of records with the following key values: 13, 7, 32, 12, 39, 5, 40, 11.

1.33 Solve the following recurrence relation by using the method of generating function: $h_n = h_{n-1} + h_{n-2}$, $n \geq 2$, $h_0 = 1$, $h_1 = 3$.

1.34 Prove that the GCD algorithm requires time logarithmic with respect to the value of the numbers and is linear with respect to the input length.

1.35 The stirling number of the second kind, $S(n, k)$, is the number of partitions of an n-element set into k classes. $S(n, k)$ is recursively defined as

$$S(n + 1, k) = S(n, k - 1) + kS(n, k)$$

with

$$S(n, k) = 0 \text{ for } k > n,$$

$$S(n, 1) = 1 \text{ for } n \geq 1,$$

$$S(0, 0) = 1,$$

$$S(n, 0) = 0 \text{ for } n \geq 1.$$

Write an algorithm to compute $S(l, r)$ for the given values of l and r where $l, r \geq 0$. Formulate the time complexity of the algorithm in terms of l and r.

1.36 Write an algorithm to compute the LCM of two given integers using the GCD algorithm of Euclid.