1. Suppose the input for a problem is \( m \) nonnegative integers, with no value higher than \( m \). Provide and justify an asymptotic bound on the size of the input in terms of \( m \).

2. Explain the average-case analysis of either QuickSort or Hash Tables (using chaining) or binary search trees.

3. Explain how Dijkstra’s algorithm or Kruskal’s algorithm or Prim’s Algorithm can be implemented so it runs in \( O((V + E)\lg(V + E)) \) time.

4. Show a trace of the Edmonds-Karp algorithm on a maximum flow problem.

5. Write the Bellman-Ford algorithm for single-source shortest paths.

6. Explain why the Bellman-Ford algorithm is correct and/or describe its running time.

7. Define one of more of: polynomial-time reduction, P complexity class, NP complexity class, NPC complexity class. What is the subset relationships between P, NP, and NPC?

8. Show that a problem is NP-complete, selected from 4-SAT-CNF, subgraph-isomorphism (p. 1100), set-partition (p. 1101), set covering (pp. 1117ff).

9. Consider solving the following problem by dynamic programming. Similar to the problem of String Matching in Chapter 32, start with a pattern string \( P \) and a text string \( T \). In this case, we want to find the best matching shift in \( T \).

   Suppose \( P = abc \). Then, of course, if the substring \( abc \) appears in \( T \), then that would be an exact (and best) match. However if \( abc \) does not appear in \( T \), but \( abzc \) appears in \( T \), that would be a kind of best match, with just one “spurious” character. Similarly, a substring \( azzbc \) within \( T \) would be a match with two spurious characters. So the problem is to find the match within \( T \) with the fewest spurious characters.

   The test will consider a particular way to define the recursive structure of this problem, turning the recursion into an efficient (polynomial-time) algorithm, and/or extracting the best match from the dynamic programming calculation.

10. For the following procedure:

    \[ \text{Weird-Random}(n) \]
    \[
    \begin{align*}
    &x \leftarrow 1 \\
    &\text{for } i \leftarrow 1 \text{ to } n \\
    &\quad r1 \leftarrow \text{Random}(1, 2) \\
    &\quad r2 \leftarrow \text{Random}(1, 2) \\
    &\quad r \leftarrow \min(r1, r2) \\
    &\quad x \leftarrow \max(x, r) \\
    &\text{return } x
    \end{align*}
    \]
(a) What is the probability that \texttt{WIERD-RANDOM(1)} returns 2?

(b) What is the probability that \texttt{WIERD-RANDOM(2)} returns 2?

(c) In terms of \( n \), what is the probability that \texttt{WIERD-RANDOM(}n\texttt{)} returns 2?