Solutions of the review problems

1. Recurrence equations for the time complexity:

\[ T(1) = c_1, \]
\[ T(n) = 2T(n-1) + c_2. \]

The solution of the recurrence equations:

We iterate the substitution operation \( i \) times and we obtain

\[ T(n) = 2^iT(n-i) + c_2(2^{i-1} + 2^{i-2} + \ldots + 2^1 + 2^0). \]

We choose \( i = n - 1 \). This gives

\[ T(n) = 2^{n-1}T(1) + c_2(2^{n-2} + 2^{n-3} + \ldots + 2^1 + 2^0). \]

We simplify the expression and we get

\[ T(n) = (c_1 + c_2)2^{n-1} - c_2. \]

The formula for \( F(n) \):

\[ F(n) = 2^{2^{n-1}}. \]

The modification of \( F(n) \):

We declare variable \( a \) as an integer. We replace the statement

\[ \text{return } (F(n-1) \ast F(n-1)) \]

by

\begin{verbatim}
begin
a:=F(n-1);
\text{return } (a \ast a)
end.
\end{verbatim}

2. The main result of chapter 9 of the textbook and the method of example 9.4 on page 302 show that \( T(n) \) is both \( O(n^2) \) and \( \Omega(n^2) \), and that \( R(n) \) is both \( O(n^{\log_2 3}) \) and \( \Omega(n^{\log_2 3}) \). Since \( \log_2 3 < 2 \), we conclude that the second algorithm has a better asymptotic performance.
3. **procedure** \( F(x: \text{elementtype}; \text{var } L: \text{LIST}); \)
   
   ```pascal```
   var
   p: position;
   begin
   p := FIRST(L);
   while p <> END(L) do
   if same(RETRIEVE(p, L), x) then
   p := END(L)
   else
   DELETE(p, L)
   end;  \{  \( F  \) \}
   ```

4. Operation PRINTLIST is both \( O(n^2) \) and \( \Omega(n^2) \), where \( n \) is the size of the input list.

5. Let \( G(n) \) be the function denoting the number of calls, which occur during the execution of \( F(n) \). A direct inspection of the tree of recursive calls of \( F \) shows that
   \[
   G(n) = G(n - 2) + G(n - 1) + 1.
   \]

   Since the executions of \( F(0) \) and of \( F(1) \) require exactly one call, therefore \( G(0) = G(1) = 1 \).

   We show the steps leading to the solution of the above non-homogeneous recurrence equation.

   **Step 1.** We look for the general solution of the homogeneous equation
   \[
   G(n) = G(n - 2) + G(n - 1).
   \]

   We assume that the solution has the form \( \alpha^n \), we plug \( \alpha^n \) into the equation, and we obtain
   \[
   \alpha^n = \alpha^{n-2} + \alpha^{n-1}.
   \]

   Dividing both sides by \( \alpha^{n-2} \) gives
   \[
   \alpha^2 = 1 + \alpha.
   \]
We solve this equation. We obtain

\[ \alpha = \frac{1+\sqrt{5}}{2} \text{ or } \alpha = \frac{1-\sqrt{5}}{2}. \]

We conclude that the general solution of the homogeneous equation has the form

\[ G(n) = c_1 \left( \frac{1+\sqrt{5}}{2} \right)^n + c_2 \left( \frac{1-\sqrt{5}}{2} \right)^n, \]

where \( c_1, c_2 \) are some constants.

Step 2. We look for a particular solution of the non-homogeneous equation

\[ G(n) = G(n-2) + G(n-1) + 1. \]

We verify directly that the constant function \( G(n) = -1 \) satisfies the equation.

Step 3. We conclude from steps 1 and 2 that the general solution of the non-homogeneous equation has the form

\[ G(n) = c_1 \left( \frac{1+\sqrt{5}}{2} \right)^n + c_2 \left( \frac{1-\sqrt{5}}{2} \right)^n - 1, \]

where \( c_1, c_2 \) are some constants.

Step 4. We compute the values of constants \( c_1, c_2 \) from equations

\[ G(0) = G(1) = 1. \]

We obtain

\[ c_1 = 1 + \frac{\sqrt{5}}{5}, \quad c_2 = 1 - \frac{\sqrt{5}}{5}. \]

Step 5. We conclude that the solution of the non-homogeneous equation has the form

\[ G(n) = \left( 1 + \frac{\sqrt{5}}{5} \right) \left( \frac{1+\sqrt{5}}{2} \right)^n + \left( 1 - \frac{\sqrt{5}}{5} \right) \left( \frac{1-\sqrt{5}}{2} \right)^n - 1. \]
The first term of the above expression has the faster rate of growth than the second term, since \[
\frac{1 + \sqrt{5}}{2} > \frac{1 - \sqrt{5}}{2}.
\]

We conclude that function \( G(n) \) has the same rate of growth as
\[
\left( \frac{1 + \frac{\sqrt{5}}{5}}{\frac{2}{2}} \right)^n.
\]

6. \( \text{function } F(n, m: \text{integer}): \text{integer}; \)

\[\begin{array}{l}
\text{var } \\
\quad K, L: \text{List}; \\
\quad k: \text{integer}; \\
\quad p: \text{position}; \\
\end{array}\]

\[\begin{array}{l}
\text{begin } \\
\quad \text{MAKENULL}(K); \\
\quad \text{INSERT}(1, \text{FIRST}(K), K); \\
\quad \text{INSERT}(1, \text{END}(K), K); \\
\quad \text{for } k := 1 \text{ to } n-1 \text{ do begin} \\
\quad \quad \text{NEXT\_LIST}(K, L); \\
\quad \quad \text{COPY}(L, K) \\
\quad \text{end}; \\
\quad p := \text{FIRST}(K); \\
\quad k := 1; \\
\quad \text{while } (k < m+1) \text{ and } (p <> \text{END}(K)) \text{ do begin} \\
\quad \quad p := \text{NEXT}(p, K); \\
\quad \quad k := k+1 \\
\quad \text{end}; \\
\quad \text{return } (\text{RETRIEVE}(p, K)) \\
\text{end}; \{F\}\]

\( \text{procedure } \text{NEXT\_LIST}( \text{K: LIST}; \text{var } L: \text{LIST}); \)

\[\begin{array}{l}
\text{var } \\
\quad p: \text{position}; \\
\quad x: \text{integer}; \\
\end{array}\]

\[\begin{array}{l}
\text{begin } \\
\quad p := \text{FIRST}(K); \\
\quad \text{MAKENULL}(L); \\
\quad \text{while } \text{NEXT}(p, K) <> \text{END}(K) \text{ do begin} \\
\quad \quad x := \text{RETRIEVE}(p,K)+\text{RETRIEVE}(\text{NEXT}(p,K),K); \\
\quad \quad \text{INSERT}(x, \text{END}(L), L); \\
\quad \quad p := \text{NEXT}(p, K) \\
\quad \text{end}; \\
\quad \text{INSERT}(1, \text{FIRST}(L), L); \\
\end{array}\]
INSERT(1, END(L), L)
end;  { NEXT_LIST }

procedure COPY( K: LIST; var L: LIST );
var
  p: position;
begin
  MAKENULL(L);
  p := FIRST(K);
  while p <> END(K) do begin
    INSERT(RETRIEVE(p, K), END(L), L);
    p := NEXT(p,K)
  end
end;  { COPY }

7.  while there are some unmarked cells do begin
  find the first unmarked cell in the post-list and mark it with L;
  find the same cell in the pre-list and mark it also with L;
  mark with I all unmarked cells of the pre-list, which occur before
  the cell you just marked with L;
  find the same cells in the post-list and also mark them with I
end;