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

$$T(n) = 2^i T(n-i) + c_2(2^{i-1} + 2^{i-2} + ... + 2^1 + 2^0) .$$

We choose $i = n-1$. This gives

$$T(n) = 2^{n-1} T(1) + c_2(2^{n-2} + 2^{n-3} + ... + 2^1 + 2^0) .$$

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

return (*F*(n-1) * *F*(n-1))

by

```
begin
        a:=F(n-1);
        return (a * a)
end.
```

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:elementtype; var L: LIST);
    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 α^n, we plug α^n into the equation, and we obtain

$$\alpha^n = \alpha^{n-2} + \alpha^{n-1}.$$

Dividing both sides by α^{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

$$c_1 = 1 + \frac{\sqrt{5}}{5},\ c_2 = 1 - \frac{\sqrt{5}}{5}.$$

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 $\left|\frac{1+\sqrt{5}}{2}\right| > \left|\frac{1-\sqrt{5}}{2}\right|$.

We conclude that function $G(n)$ has the same rate of growth as

$$\left(1+\frac{\sqrt{5}}{5}\right)\left(\frac{1+\sqrt{5}}{2}\right)^n.$$

6.
```
function F(n, m: integer): integer;
    var
        K, L: List;
        k: integer;
        p: position;
    begin
        MAKENULL(K);
        INSERT(1, FIRST(K), K);
        INSERT(1, END(K), K);
        for k := 1 to n-1 do begin
            NEXT_LIST(K, L);
            COPY(L, K)
        end;
        p := FIRST(K);
        k := 1;
        while (k < m+1) and (p <> END(K)) do begin
            p := NEXT(p, K);
            k := k+1
        end;
        return (RETRIEVE(p, K))
    end; {F}

procedure NEXT_LIST( K: LIST; var L: LIST);
    var
        p: position;
        x: integer;
    begin
        p := FIRST(K);
        MAKENULL(L);
        while NEXT(p, K) <> END(K) do begin
            x := RETRIEVE(p,K)+RETRIEVE(NEXT(p,K),K);
            INSERT(x, END(L), L);
            p := NEXT(p, K)
        end;
        INSERT(1, FIRST(L), L);
```

```
            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;
```