

### Ćwiczenia 3

*Ćwiczenie 10.* Algorytm wyznaczający  $n$ -ty wyraz ciągu Fibonacciego technik programowania dynamicznego bez użycia tablicy.

```
FIB( $n$ )  
   $F0 = 0$   
   $F1 = 1$   
  for  $i = 2$  to  $n$   
    do  $Fk = F0 + F1$   
       $F0 = F1$   
       $F1 = Fk$   
  return  $Fk$ 
```

*Ćwiczenie 11.* Dana jest tablica  $A[1..n]$  zawierająca liczby. Podaj algorytm wyznaczający liczbę najmniejszych elementów w tej tablicy. Wolno tylko raz przeglądać tablicę.

```
L-MIN( $n, A$ )  
   $min = A[1]$   
   $k = 1$   
  for  $i = 2$  to  $n$   
    do if  $A[i] < min$   
      then  $min = A[i]$   
         $k = 1$   
    else if  $A[i] == min$   
      then  $k = k + 1$   
  return  $k$ 
```

*Ćwiczenie 12.* Algorytm wyznaczający największą wartość wektora  $A[1..n]$  oraz wszystkie pozycje (wartości indeksów), na których się ona znajduje. Tablicę  $A$  wolno przeglądać tylko raz.

MAX-POZ( $A, n$ )

$poz = 1$

$kon = 1$

$B[1] = \text{true}$

**for**  $i = 2$  **to**  $n$

**do if**  $A[i] > A[poz]$

**then**  $poz = i$

$kon = i$

$B[i] = \text{true}$

**else if**  $A[i] == A[poz]$

**then**  $kon = i$

$B[i] = \text{true}$

**else**  $B[i] = \text{false}$

    pisz ( $A[poz]$ )

**for**  $i = poz$  **to**  $kon$

**do if**  $B[i]$

**then** pisz ( $i$ )

*Ćwiczenie 13.* Dana jest tablica  $A[1..n, 1..n]$  zawierająca liczby. Algorytm wyznaczający sumę elementów leżących "na brzegu" tej tablicy, tj. w pierwszym i ostatnim wierszu oraz w pierwszej i ostatniej kolumnie:

a) cztery pętle for

$s = 0$

**for**  $i = 1$  **to**  $n$

**do**  $s = s + A[1, i]$

**for**  $i = 1$  **to**  $n$

**do**  $s = s + A[n, i]$

**for**  $i = 2$  **to**  $n - 1$

**do**  $s = s + A[i, 1]$

**for**  $i = 2$  **to**  $n - 1$

**do**  $s = s + A[i, n]$

b) dwie pętle for

$s = 0$

**for**  $i = 1$  **to**  $n$

**do**  $s = s + A[1, i] + A[n, i]$

**for**  $i = 2$  **to**  $n - 1$

**do**  $s = s + A[i, 1] + A[i, n]$

c) jedna pętla for

$s = A[1, 1] + A[1, n] + A[n, 1] + A[n, n]$

**for**  $i = 2$  **to**  $n - 1$

**do**  $s = s + A[i, 1] + A[1, i] + A[i, n] + A[n, i]$

*Ćwiczenie 14.* Dana jest tablica  $A[1..m, 1..n]$  zawierająca liczby. Algorytm wyznaczający najmniejszy i największy element tej tablicy oraz ich miejsca pierwszego pojawienia się podczas jednokrotnego przeszukiwania tablicy.

```
 $min = A[1, 1]$   
 $kmin = 1$   
 $lmin = 1$   
 $max = A[1, 1]$   
 $kmax = 1$   
 $lmax = 1$   
for  $i = 1$  to  $m$   
  do for  $j = 1$  to  $n$   
    do if  $A[i, j] > max$   
      then  $max = A[i, j]$   
         $kmax = i$   
         $lmax = j$   
    else if  $A[i, j] < min$   
      then  $min = A[i, j]$   
         $kmin = i$   
         $lmin = j$   
return  $max, kmax, lmax, min, kmin, lmin$ 
```