Seminar 2 – Tablouri si structuri

13.10.2020

Pr 1.

Scrieti o functie succesor peste un tip Data calendaristica

Exemple:

13.10.2020 🡪 14.10.2020

31.10.2020 🡪 01.11.2020

31.12.2020 🡪 01.01.2021

28.02.2020 🡪 29.02.2020

28.02.1900 🡪 01.03.1900

struct TData {

int zz

int ll

int aa

}

function isLeapYear(int an)

begin

return (an % 400 == 0) or ((an % 4 == 0) and (an % 100 != 0))

end

// 31 28/29 31 30 31 30 31 31 30 31 30 31

function lastDayofMonth(TData data)

begin

if (data.ll == 2) then

if (isLeapYear(data.aa)) then return 29

else return 28

if ( data.ll == 4 or data.ll == 6 or data.ll == 9 or data.ll == 11 ) then return 30

return 31

end

function succesor(TData data)

begin

if (data.zz != lastDayofMonth(data)) then data.zz <- data.zz + 1

else { data.zz <- 1; data.ll <- (data.ll % 12) + 1; if (data.ll == 1) then data.aa <- data.aa + 1}

return data

end

Pr. 2

Input: n >= 1, a un vector de numere intregi

Output: sa se gaseasca cea mai lunga secventa de elemente consecutive din a a.i. suma secventei sa fie maxima.

max {

i = 0, .., n-1

j = i, …, n-1

sum(a[i] ,… a[j]) // b[j] – b[i-1] ~ n^2

}

b[j] = a[0] + a[1] + … +a[j]

~ BF: n^3 pasi

Consideram toate j+1 secvente care se termina in pozitia j (0 <= j<= n-1)

sumele acestora

argmax (b[j] – b[i]) , i = 0,…, j este echivalenta

b[j] – min(b[i]), I =0,…j

Exemple:

n = 7, a = (-3, -5, -1, -100, -6, -2, -12)

0 1 2 3 4 5 6 7 8

n = 9, a = (2, 3, -4, -2, 9, -2, 1, 7, -3) (4, 4, 15)

b =(2, 5, 1, -1, 8, 6, 7, 14, 11)

Reprezentarea solutiei: pi, pf, suma

… (n, a[0..n-1])s

b[0] <- a[0]

for I <- 1 to n-1 do b[i] <- b[i-1] + a[i] // n pasi

pi <- 0; pf <- 0; suma <- a[0]; min <- b[0]; pmin <- 0

for j = 1 to n-1 to { // ~ n pasi

if b[j] < min then {min <- b[j]; pmin <- j}

if (b[j] –min > suma) then {pi <- pmin+1 ; pf <- j; suma <- b[j] –min}

}

~ n pasi

TEMA: scrieti o procedura in pseudocod

implementati algoritmul

Pr. 3

Input: n >=1, a[0..n-1] tablou cu n elemente nr intregi

Output: true daca exista un element majoritar (cu nr de aparitii >= n/2 + 1)

false, caz contrar

Exemple:

n =5, a = (1, 2 ,2 , 1, 1) 🡪 true

n =5, a = (1, 2 ,2 , 3, 1) 🡪 false

1. I = 0, .., n/2
   1. numaram frecventa lui a[i]
   2. testam daca aceast fr. >= n/2 + 1
2. sortam ( ~ n log n pasi)
3. anihilare de perechi de numere distincte + frecventa candidatului ramas (n + n ~ n pasi )

function majoritar( n, a[0..n-1))

begin

// (1,1,1,2,2)

candidat <- a[0]; nr <- 1

for i <- 1 to n-1 do {

if … ? TEMA !

fr <- 0

for I <- 0 to n-1 do if a[i] == candidat then fr <- fr + 1

return fr > n/2

end