

# Examen parcial de Programació 2020-2021

8/4/2021

Durada: 1:40

## Exercici 1: Centres d'ordre (4 punts)

Sigui  $A_{n \times n}$  una matriu quadrada d'enters positius amb  $n \geq 3$ . Es diu que una component  $A[i, j]$  es un  $k$ -centre d'ordre si la submatriu  $A[(i - k) : (i + k), (j - k) : (j + k)]$  és ordenada per files en forma creixent. Per exemple, si tenim la següent matriu  $A$ :

	1	2	3	4	5	6	7	8	9
1	11	12	13	2	1	3	1	1	1
2	14	15	16	17	18	19	19	1	1
3	17	18	19	20	21	25	27	1	1
4	2	1	27	30	40	45	46	1	1
5	3	11	47	50	52	63	65	1	1
6	90	65	70	71	74	76	80	1	1
7	12	74	75	76	12	10	10	1	1
8	11	81	90	95	13	11	10	1	1
9	77	96	96	97	15	11	14	1	1

Podem veure que els elements dins d'un cercle com, per exemple  $A[2, 2]$ ,  $A[4, 4]$ ,  $A[4, 5]$  o  $A[8, 3]$ , són 1-centres d'ordre. En particular,  $A[4, 5]$  és també un 2-centre d'ordre.

Es demana que facis una funció `numCentres` que, donada una matriu  $A_{n \times n}$ ,  $n \geq 3$  i un número  $0 < k \leq (n-2)$ , escrigui una llista amb els parells  $(i, j)$  corresponent a les posicions de la matriu  $A$  que contenen  $k$ -centres d'ordre.

Per exemple, la crida `numCentres(A, 1)` ha d'escriure els parells  $(2, 2)$ ,  $(3, 4)$ ,  $(3, 5)$ ,  $(3, 6)$ ,  $(4, 4)$ ,  $(4, 5)$ ,  $(4, 6)$ ,  $(5, 4)$ ,  $(5, 5)$ ,  $(5, 6)$ ,  $(6, 3)$ ,  $(7, 3)$ ,  $(8, 3)$ . La crida `numCentres(A, 2)` ha d'escriure  $(4, 5)$ . **Si  $A$  no té cap  $k$ -centre d'ordre, la funció no ha d'escriure res.**

**Assumeix que disposes d'una funció `ordenada(M)` que, donada una matriu  $M$ , torna `TRUE` si la matriu està ordenada per files i `FALSE` en cas contrari.** Una matriu és ordenada en forma creixent per files si recorrent-la per files s'obté una llista en ordre ascendent.

## Exercici 2: Què calcula? (3 punts)

Tenim les següents funcions ja implementades:

```
1 p ← function(v,e){
2   k ← length(v)
3   while (k ≥ 1 && e < v[k]){
4     v[k+1] ← v[k]
5     k ← k-1
6   }
7   v[k+1] ← e
8   return(v)
9 }
10
11 quefaig ← function(a){
12   r ← c()
13   for (i in 1: nrow(a)){
14     t ← FALSE
15     j ← 1
16     while (!t && j ≤ ncol(a)){
17       if (a[i,j] > a[i,1]){
18         t ← TRUE
19         r ← p(r,a[i,j])
20       }
21       j ← j+1
22     }
23     if (j > ncol(a)) r ← p(r,a[i,1])
24   }
25   return(r)
26 }
```

1. Si la matriu d'entrada d'aquesta funció és la següent:

```
> m1
  [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,]   1   2  10   3   4   5   7  20
[2,]   2   1   2   1   3   4   5   7
[3,]  20  20   1   2  10  30   4   5
[4,]   7  20  23   1   2  10   3   4
[5,]   5   7  20  23   1   2  10   3
```

Quina serà la sortida de `quefaig(m1)`?

2. I si la matriu d'entrada d'aquesta funció és la següent:

```
> m2
  [,1]
[1,]   1
[2,]  20
[3,]   3
[4,]   4
[5,]  51
```

Quina serà la sortida de `quefaig(m2)`?

### Exercici 3 (3 punts)

Disposem de dos list L1 i L2 on tenim els *stocks* corresponents de dues botigues de BonPreu. En tots dos lists tenim la quantitat de producte que tenim en stock de cadascun dels productes (identificats pel nom del producte).

Es demana que facis una funció que donats els dos lists (L1 i L2) tal i com s'han descrit, modifiqui el list L1 per a què contingui l'stock acumulat (la suma dels dos stocks). Pot ser que hi hagi elements en L2 que no estan en L1 i viceversa.

Si tenim per exemple:

```
L1 <- list (llets = 10, platans = 6, desodorants = 3,
           tomaquets = 6, kellogs = 25, patates = 30)
L2 <- list (tomaquets = 9, kellogs = 10, llets = 23,
           aigues = 15, llimones = 7, desodorant = 5,
           patates = 18)
```

el resultat ha de ser:

```
L1 <- list (llets = 33, platans = 6, desodorants = 8,
           tomaquets = 15, kellogs = 35, patates = 48,
           aigues = 15, llimones = 7)
```