

Nom i cognoms:

## Algorísmia i Programació 1, GCED, 28 de juny de 2018

*L'examen dura tres hores. Es valorarà la concisió, claredat i brevetat a més de la completesa i l'exactitud de les respostes. Contesteu a l'espai marcat amb bona lletra. Poseu el vostre nom a cada full. No podeu consultar cap material addicional.*

### 1 Pupurri

(2'5 punts)

1. Expliqueu què és un *intèrpret* (en informàtica).

2. Expliqueu què calcula l'algorisme d'Euclides.

3. Expliqueu què és una *precondició*.

4. Anomeneu quatre algorismes d'ordenació diferents vistos al curs d'AP1. Subratlleu el(s) més ràpid(s) de tots ells (en el cas pitjor).

5. Expliqueu què és necessari per poder fer una cerca binària en un vector.

6. Declareu, en C++, una variable  $M$  que representi una matriu de  $n \times n$  reals, tots inicialitzats a zero.

7. Considereu el programa següent en Python3:

```
def f(n):  
    if n == 4:  
        return n  
    return 2 * f(n + 1)
```

Què retorna  $f(2)$ ?

Què retorna  $f(8)$ ?

8. Digueu què retorna  $g(x, y)$  quan  $x, y \geq 0$  amb aquesta funció en Python3:

```
def g(x, y):  
    if x == 0:  
        return y  
    else:  
        return g(x - 1, x + y)
```

9. Digueu què escriu el fragment següent en Python3:

```
a = [1, 2]  
b = [a, a]  
b[0][0] = 9  
print(b)
```

10. Aquesta funció hauria d'ordenar llistes, però té un *bug*. Trobeu-lo i esmeneu-lo.

```
def sort(L):  
    for i in range(1, len(L)):  
        j = i - 1  
        k = L[i]  
        while j >= 0 and L[j] > k:  
            L[j+1] = L[j]  
            j = j - 1  
        L[j] = k
```

## 2 Classificar elements

(2'5 punts)

Considereu la funció

```
void classificarMod3 (const vector<int>& V);
```

amb l'especificació següent:

$V$  és un vector que conté enters positius. La funció reordena els elements del vector en tres segments en funció del residu en ser dividits per 3, de tal manera que els elements amb residu 0 (mod 3) queden a les posicions més baixes i els elements amb residu 2 (mod 3) queden a les posicions més altes. La funció no pot fer servir cap vector auxiliar.

Per exemple, si l'entrada és el vector  $V = [8\ 1\ 4\ 6\ 5\ 7\ 9\ 8\ 3\ 5\ 4]$ , la sortida podria ser el vector  $V = [9\ 6\ 3\ |\ 1\ 4\ 7\ 4\ |\ 8\ 5\ 8]$ . Dins de cada segment, els elements poden estar en qualsevol ordre.

a) Dissenyeu un invariant per al bucle principal de la funció. L'invariant haurà d'explicar el significat de cadascuna de les variables que intervenen en el bucle principal i les propietats que compleix. L'invariant *no* ha d'explicar com funciona l'algorisme. Es recomana representar l'invariant gràficament amb un dibuix del vector.

b) Implementeu la funció *classificarMod3* segons l'invariant proposat i indiqueu la complexitat del vostre algorisme.

### 3 Punt fix

(2'5 punts)

Dissenyeu la funció `int puntFix (const vector<int>& A)`; on  $A$  és un vector d'enters en ordre estrictament creixent. La funció retorna l'índex d'un element  $i$  tal que  $A[i] = i$ . En cas que existeixi més d'un element que compleixi la propietat, la funció pot retornar qualsevol d'ells. En cas que no n'existeixi cap, la funció retorna -1.

Cal que el disseny sigui *eficient* i *recursiu*, és a dir, no es pot fer servir cap instrucció iterativa del tipus **while** o **for**. Es recomana dissenyar una funció auxiliar que implementi la recursivitat.

Digueu quina és la complexitat de l'algorisme implementat.

## 4 Trobar la fila escalada

(2'5 punts)

Considereu les definicions següents:

```
using Fila = vector<double>;  
using Matriu = vector<Fila>;  
int trobarFilaEscalada(const Matriu& A, const Fila& F);
```

La funció *trobarFilaEscalada* cerca una fila  $A[i]$  que sigui un escalament d' $F$ , és a dir,  $A[i] = c \cdot F$  per algun escalar  $c \neq 0$ . En cas que trobi una fila  $A[i]$  que compleixi la propietat, la funció retorna  $i$ . En cas que n'hi hagi més d'una, retorna la  $i$  més petita. En cas que no trobi cap fila, retorna -1.

Per exemple, si

$$A = \begin{bmatrix} 3 & 4 & 5 & 1 & 7 \\ 6 & 3 & 1 & 5 & 6 \\ 0 & 2 & 3 & -1 & 4 \\ 4 & 0 & -3 & 8 & 9 \end{bmatrix} \quad \text{i} \quad F = [0 \quad 4 \quad 6 \quad -2 \quad 8]$$

llavors la crida *trobarFilaEscalada*( $A, F$ ) hauria de retornar 2, atès que  $A[2] = 1/2 \cdot F$ .

Podeu suposar que el nombre d'elements del vector és igual al nombre de columnes de la matriu. Podeu ignorar els problemes de precisió amb els nombres reals. Es valorarà principalment l'eficiència de la funció i la seva senzillesa.

**Consell:** Cal anar en compte amb la presència de zeros, tant en el vector com en la matriu.