PRG - ETSInf. TEORIA. Curs 2017-18. Parcial 1. Recuperació
19 de juny de 2018. Durada: 2 hores.

Nota: L'examen s'avalua sobre 10 punts, però el seu pes específic en la nota final de PRG és **de 3 punts**.

1. $\boxed{\text{4 punts}}$ Siga `a` un array de `double`, on les seues components representen valors de la coordenada Y d'una enumeració de punts pertanyents a una recta de pendent positiva, de manera que l'array està ordenat ascendentment.

   S'ha d'escriure un mètode recursiu que cerque el punt de tall de la recta amb l'eix X, o que retorne `-1` si el punt de tall no es troba entre els d'`a`.

   Per exemple: si l'array és {`-7.4, -1.3, 0.0, 1.8, 2.3, 3.6`}, el mètode ha de retornar `2`. Si l'array és {`-7.4, -1.3, 1.8, 2.6, 3.6`}, el mètode ha de retornar `-1`.

   Es demana:

   a) (0.75 punts) Perfil del mètode, amb els paràmetres adequats per a resoldre recursivament el problema, i precondició relativa a aquests paràmetres.
   b) (1.25 punts) Cas base i cas general.
   c) (1.50 punts) Implementació en Java.
   d) (0.50 punts) Crida inicial perquè, donat un cert array `a`, es realitze el càlcul sobre tot l'array.

Solució:

1. Una possible solució consisteix a definir un mètode amb el següent perfil, i que segueix una estratègia de cerca dicotòmica:

```
/** Cerca el valor 0 en a[left..right]. Precondició: 0 <= left, right < a.length,
 *  a[left..right] ordenat ascendentment.  */
public static int intercepts(double[] a, int left, int right) {
    if (left > right) { return -1; }
    else {
        int middle = (left + right) / 2;
        if (a[middle] == 0) { return middle; }
        else if (a[middle] > 0) {
            return intercepts(a, left, middle - 1);
        } else {
            return intercepts(a, middle + 1, right); }
    }
}
```

   La crida inicial per a resoldre el problema sobre un cert array `a` hauria de ser `intercepts(a, 0, a.length - 1)`.

2. Una altra possible solució consisteix a definir un mètode amb el següent perfil, i que segueix una estratègia de cerca lineal:

```
/** Cerca el valor 0 en a[ini..a.length - 1].
 *  Precondició: 0 <= ini, a[ini..a.length - 1] ordenat ascendentment. */
public static int intercepts(double[] a, int ini) {
    if (ini >= a.length) { return -1; }
    else if (a[ini] == 0) { return ini; }
    else if (a[ini] > 0) {return -1;}
    else { return intercepts(a, ini + 1); }
}
```

   La crida inicial per a resoldre el problema sobre un cert array `a` hauria de ser `intercepts(a, 0)`.

Ambdós mètodes són $\Omega(1)$, encara que el primer és $O(logn)$, mentre que el segon és $O(n)$, sent $n =$ `a.length`.

2. $\boxed{\text{3 punts}}$ Donada una matriu quadrada `m` de caràcters i un caràcter `c`, el següent mètode escriu les paraules o seqüències de caràcters que apareixen en cada fila, eliminant d'elles cada aparició de `c`.

```
/** Precondició: m és una matriu quadrada. */
public static void escriuSense(char[][] m, char c) {
    int dim = m.length;
    for (int i = 0; i < dim; i++) {
        for (int j = 0; j < dim; j++) {
            if (m[i][j] != c) {
                System.out.print(m[i][j]);
            }
        }
        System.out.println();
    }
}
```

Per exemple, si `m` = {{'e','e','l','e'}, {'m','e','m','e'}, {'n','u','l','l'}, {'c','a','s','e'}}, i `c` = 'e', aleshores el mètode escriu:

```
l
mm
null
cas
```

Es demana:

a) (0.25 punts) Indiqueu quina és la grandària o talla del problema, així com l'expressió que la representa.

b) (0.75 punts) Indiqueu, i justifiqueu, si existeixen diferents instàncies significatives per al cost temporal de l'algorisme i identifiqueu-les si és el cas.

c) (1.50 punts) Trieu una unitat de mesura per a l'estimació del cost (passos de programa, instrucció crítica) i d'acord amb ella obteniu una expressió matemàtica, el més precisa possible, del cost temporal del mètode, distingint el cost de les instàncies més significatives en cas d'haver-les.

d) (0.50 punts) Expresseu el resultat anterior utilitzant notació asimptòtica.

Solució:

a) La talla del problema és la dimensió de la matriu `m` i l'expressió que la representa és `m.length`. D'ara endavant, anomenarem a aquest nombre n. Açò és, $n =$ `m.length`.

b) No existeixen diferents instàncies. El mètode examina tots els caràcters de totes les files de `m`.

c) Si escollim com unitat de mesura el pas de programa, tenim:

$T(n) = 1 + \sum_{i=0}^{n-1}(1 + \sum_{j=0}^{n-1} 1) = 1 + \sum_{i=0}^{n-1}(1 + n) = 1 + n + n^2 \; p.p.$

Si escollim com unitat de mesura la instrucció crítica i considerant com a tal, per exemple, l'avaluació de la condició `m[i][j] != c` (de cost unitari), tenim:

$T(n) = \sum_{i=0}^{n-1}(\sum_{j=0}^{n-1} 1) = \sum_{i=0}^{n-1} n = n^2 \; i.c.$, és a dir, $n^2 \; p.p.$ menyspreant termes d'ordre inferior.

d) En notació asimptòtica $T(n) \in \Theta(n^2)$.

3. $\boxed{\text{3 punts}}$ Es desitja calcular el cost del següent mètode recursiu, que donats $\mathtt{x} > 1$ i $\mathtt{d} \leq \mathtt{x}$, comprova si `x` no té cap divisor propi en el rang [`d`, `x`[:

```
/** Precondició: x > 1 && 1 < d <= x */
public static boolean senseDivisors(int x, int d) {
    if (x == d) { return true; }
    else {
        if (x % d == 0) { return false; }
        else { return senseDivisors(x, d + 1); }
    }
}
```

Es demana:

a) (0.25 punts) Indiqueu quina és la talla o grandària del problema, així com l'expressió que la representa.

b) (0.75 punts) Indiqueu, i justifiqueu, si existeixen diferents instàncies significatives per al cost temporal de l'algorisme i identifiqueu-les si és el cas.

c) (1.50 punts) Escriviu l'equació de recurrència del cost temporal en funció de la talla per a cadascun dels casos si hi ha més d'un, o una única equació si solament hi haguera un cas. Ha de resoldre's per substitució.

d) (0.25 punts) Expresseu el resultat anterior utilitzant notació asimptòtica.

e) (0.25 punts) Quin seria el cost asimptòtic en funció del valor de `x` de la crida `senseDivisors(x, 2)`, és a dir, d'averiguar si `x` és primer?

Solució:

a) La talla del problema és la diferència entre els valors dels arguments `x - d`. Anomenarem n a aquest valor d'ara endavant.

b) Es tracta d'una cerca del primer divisor propi de `x` major o igual que `d` i, per tant, hi ha instàncies significatives. En el cas millor, `d` és divisor de `x`. En el cas pitjor, `x` no té divisors propis majors o iguals que `d`.

c) Considerant el cost expressat en passos de programa, en el cas millor $T^m(n) = 1$ *p.p.* En el cas pitjor, plantegem l'equació de recurrència considerant les talles corresponents al cas base i al cas general:

$$T^p(n) = \begin{cases} 1 + T^p(n-1) & \text{si } n > 0 \quad (\text{quan } \mathtt{x} > \mathtt{d}) \\ 1 & \text{si } n = 0 \quad (\text{quan } \mathtt{x} = \mathtt{d}) \end{cases}$$

Resolent per substitució:

$T^p(n) = 1 + T^p(n-1) = 2 + T^p(n-2) = 3 + T^p(n-3) = \ldots = k + T^p(n-k)$ després de k passos de substitució. S'arriba al cas base $T^p(0) = 1$ després de $k = n$ passos de substitució, amb el que $T^p(n) = n + 1$ *p.p.*

d) En notació asimptòtica, el cost temporal és: $T^m(n) \in \Theta(1)$, $T^p(n) \in \Theta(n)$. És a dir, $T(n) \in \Omega(1)$ i $T(n) \in O(n)$.

e) En el cas particular de la crida `senseDivisors(x, 2)`, la talla és $n = \mathtt{x} - 2$ i, per tant, el cost en funció de `x` és $\Omega(1), O(\mathtt{x})$.