Recuperació Primer Parcial de PRG - ETSInf
Data: 19 de juny de 2012. Duració: 2 hores

1. (2 punts) Siga `a` un array d'`int`, i `x` un `int`. Es demana un mètode recursiu que indique si els elements d'`a` formen una progressió geomètrica de raó `x`, és a dir, si cada component `a[i+1]` de l'array val `a[i]*x`. Per exemple, per a `a = {3,6,12,24,48}` i x=2 el mètode ha de retornar `true`, per a `a = {3,6,12,33,48}` i x=2 el mètode ha de retornar `false`. Si sols hi ha un element, s'entén que és una progressió geomètrica siga quin siga el valor de `x`.

   Indicar quina haurà de ser la primera crida.

Solució:

```
/** Comprova si les components d'a[i..a.length-1], 0<=i<a.lenght,
 *  formen una progressió geomètrica de raó x
 */
public static boolean geometrica(int[] a,int i,int x){
   if (i==a.length-1) return true;
   else { if (a[i+1]==a[i]*x) return geometrica(a,i+1,x);
          else return false;
        }
}
```

La primera crida hauria de ser `geometrica(a,0,x)`.

2. (2 punts) Escriure un mètode recursiu que donat un enter $n \geq 0$, escriga en la sortida estàndard, i en la mateixa línia, els valors $-n$ $-(n-1)\ldots$ `-2 -1 0 1 2` $\ldots(n-1)$ n. Per exemple, per a $n = 3$, en la sortida s'ha d'escriure:

```
-3 -2 -1 0 1 2 3
```

Solució:

```
/** n>=0 */
public static void escriu(int n) {
  if (n==0) System.out.print(0);
  else { System.out.print(-n + " ");
         escriu(n-1);
         System.out.print(" " + n);
       }
}
```

3. (2.5 punts) Donat el següent mètode:

```
/** n>=0, 1<=x<=9 */
public static boolean cercarX(int n,int x){
   if (n>0)
       if (n%10==x) return true;
```

```
        else return cercarX(n/10,x);
    else return false;
}
```

Es demana:

a) Indicar quina és la talla del problema i quina expressió la defineix.

b) Determinar si existeixen instàncies significatives. Si n'hi ha, identificar les que representen els casos millor i pitjor de l'algorisme.

c) Escriure l'equació de recurrència del cost temporal en funció de la talla per a cada un dels casos si n'hi ha diversos, o una única equació si només hi hagués un cas. Cal resoldre-la per substitució.

d) Expressar el resultat anterior fent servir notació asimptòtica.

Solució:

a) La talla del problema és el valor de l'argument del mètode, açò és, `n`.

b) Sí que hi ha instàncies significatives, és una cerca. En el millor cas, `x` és la xifra de les unitats de `n`, i en el cas pitjor cap xifra de `n` és `x`.

c) En el cas millor $T^m(n) \in \Theta(1)$.

En el cas pitjor, l'equació de recurrència per al cost, en passos de programa, és:

$$T^p(n) = \begin{cases} T^p(n/10) + 1 & \text{si } n > 0 \\ 1 & \text{si } n = 0 \end{cases}$$

Resolent-la per substitució:

$T^p(n) = T^p(n/10) + 1 = T^p(n/10^2) + 2 = \ldots = T^p(n/10^i) + i$. Quan $i = 1 + \lfloor log_{10} n \rfloor$ s'arriba al cas base en el que $T^p(0) = 1$.

Amb el que $T^p(n) = 2 + \lfloor log_{10} n \rfloor$.

d) En notació asimptòtica: $T^m(n) \in \Theta(1)$ i $T^p(n) \in \Theta(\log n)$, és a dir, les cotes per al cost són $T(n) \in \Omega(1)$ i $T(n) \in O(\log n)$.

4. (3.5 punts) Siga `a` un array $\{\mathtt{a}_0, \mathtt{a}_1, \mathtt{a}_2, \ldots, \mathtt{a}_{n-1}\}$ de `double`, que representa els coeficients d'un polinomi $a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$.

Per tal de calcular el valor del polinomi per a un x donat, es proposen els següents mètodes:

Mètode 1:

```
/** a.length>=1 */
public static double polinomi1(double[] a, double x) {
    double result = a[0];
    for(int i=1; i<a.length; i++){
        double pot = 1;
        for(int k=1; k<=i; k++) pot = pot*x;
        result += a[i]*pot;
    }
    return result;
}
```

Mètode 2:

```
/** a.length>=1 */
public static double polinomi2(double[] a, double x) {
    double result = a[a.length-1];
    for(int i=a.length-2; i>=0; i--)
        result = result*x + a[i];
    return result;
}
```

Es demana:

a) **Per a cada mètode:**

1. Indicar quina és la grandària o talla del problema, així com l'expressió que la representa.
2. Identificar, en cas que n'hi hagués, les instàncies del problema que representen el cas millor i pitjor de l'algorisme.
3. Triar una unitat de mesura per a l'estimació del cost (passos de programa, instrucció crítica) i d'acord amb ella, obtindre una expressió matemàtica, el més precisa possible, del cost temporal del programa, a nivell global o en les instàncies més significatives si les hi ha.
4. Expressar el resultat anterior utilitzant notació asimptòtica.

b) Indicar quin dels dos mètodes és més eficient i perquè.

Solució:

a) **Mètode 1.**

1) La talla $n =$`a.length`, és a dir, el nombre d'elements d'`a`.
2) No hi ha diferents instàncies, per lo que sols s'estudiarà una funció $T(n)$, vàlida per a qualsevol entrada de talla n.
3) En passos de programa,

$$T(n) = 1 + \sum_{i=1}^{n-1}(1+i) = 1 + (2 + 3 + \ldots + n) = \frac{(n+1)\cdot n}{2} \in \Theta(n^2)$$

4) En resum, $T(n) \in \Theta(n^2)$.

Mètode 2.

1) Ídem que per al mètode 1.
2) Ídem que per al mètode 1.
3) En passos de programa,

$$T(n) = 1 + \sum_{i=1}^{n-1} 1 = n \in \Theta(n)$$

4) En resum, $T(n) \in \Theta(n)$.

b) El primer mètode és quadràtic amb la talla del problema i el segon és linial, per lo que es pot concloure que el segon és més eficient.

De l'anàlisi realitzat s'observa que el primer mètode recalcula completament la potència de `x` elevat a `i` en cada passada del bucle, operació que va resultant més costosa a mesura que augmenta `i`. El segon mètode ho evita fent que les potències de `x` acumulades en `result` vagen augmentant el seu grau en 1 en cada passada del bucle.