Recuperació parcial 1 - Pràctiques - PRG - ETSInf - Curs 2012/13.
17 de juny de 2013. Duració: 50 minuts.

NOM I GRUP DE PRÀCTIQUES:

1. 2 punts Considera el següent algorisme vist en pràctiques:

```
public static void hanoi(int n, char org, char dest, char aux) {
    if (n==1) System.out.println("Mou disc des de " + org + " a " + dest);
    else {
        hanoi(n-1,org,aux,dest);
        System.out.println("Mou disc des de " + org + " a " + dest);
        hanoi(n-1,aux,dest,org);
    }
}
```

Contesta a les següents qüestions:

(a) Sabent que en el cas d'una torre de 10 discos l'algorisme tarda 10 mil·lisegons, quants tardaria si la torre tinguera 11 discos?

(b) I si tinguera 12?

Solució:

(a) El temps de `hanoi` és exponencial, $T(n) \in \Theta(2^n)$ i per tant $T(n) = k \times 2^n$. El valor de la constant k es pot conèixer a partir de la dada de l'enunciat $T(10) = k \times 2^{10} = 10$, $k = 10 \times 2^{-10}$. Així, si el número de discos és 11 es té que $T(11) = k \times 2^{11} = 20$ mil·lisegons.

(b) Si el número de discos és 12 es té que $T(12) = k \times 2^{12} = 40$ mil·lisegons.

2. 3 punts Implementar un mètode **recursiu** amb el següent perfil:

```
public static boolean esSufixe(String a, String b)
```

que comprove si la `String a` és sufixe de la `String b`. Una cadena de caràcters és sufixe d'un altra cadena si tots els seus caràcters apareixen, en el mateix ordre, en les últimes posicions de la segona cadena. Es considera que una cadena buida és sufixe de qualsevol cadena.

NOTA: En la resolució d'aquest exercici **només** es poden usar els següents mètodes de la classe `String`:

- `s.charAt(i)` torna el caràcter que ocupa la posició `i` en `s`.
- `s.substring(i,j)` torna una subcadena de `s` formada pels caràcters compresos entre les posicions `i` i `j-1`.
- `s.length()` torna la longitud de `s`.

Solució:

```
public static boolean esSufixe(String a, String b) {
      if (a.length()==0) return true;
      else if (a.length()>b.length()) return false;
      else return a.charAt(a.length()-1)==b.charAt(b.length()-1) &&
                  esSufixe(a.substring(0,a.length()-1),
                           b.substring(0,b.length()-1));
}
```

3. $\boxed{\text{5 punts}}$ Considera la taula següent amb els temps d'execució de dos algorismes expressats en mil·lisegons:

```
# Talla       Alg. A      Alg. B
#--------------------------------
  5000        28.403      18.526
 10000       112.171      75.771
 15000       252.470     160.113
 20000       448.906     289.250
 25000       701.659     447.238
 30000      1010.749     667.121
 35000      1375.908     914.553
 40000      1797.765    1160.216
 45000      2277.439    1523.284
 50000      2806.755    1835.431
```

Contesta a les següents qüestions:

(a) Quin és el cost asimptòtic que s'ajusta millor a les dades de la columna `Alg. A`?
(b) Quin és el cost asimptòtic que s'ajusta millor a les dades de la columna `Alg. B`?
(c) Escriu una estimació del temps **en segons** que utilitzaria cada algorisme per a una talla de 150000.
(d) Quin algorisme té un comportament asimptòtic millor? Quin utilitzaries en la pràctica?

Solució:

(a) El cost asimptòtic que s'ajusta millor a les dades de la columna `Alg. A` és un cost quadràtic. Si anomenem n a la talla del problema, $T(n) \in \Theta(n^2)$. Es pot comprovar que quan la talla es duplica (de n a $2n$), el temps es multiplica per 4 (de n^2 a 2^2n^2). Per exemple, per a $n = 20000$ el temps és 448.906 mil·lisegons i per a $n = 40000$ el temps és 1797.765 mil·lisegons, aproximadament 4 vegades el de $n = 20000$.

(b) El cost asimptòtic que s'ajusta millor a les dades de la columna `Alg. B` també és un cost quadràtic. Per exemple, per a $n = 20000$ el temps és 289.250 mil·lisegons i per a $n = 40000$ el temps és 1160.216 mil·lisegons, aproximadament 4 vegades el de $n = 20000$.

(c) Per a una talla de 150000 elements, el temps d'execució del `Alg. A` seria aproximadament $10^2 \times 252.470 \approx 25000$ mil·lisegons $= 25$ segons.

El temps d'execució del `Alg. B`, per a una talla de 150000 elements, seria aproximadament $10^2 \times 160.113 \approx 16000$ mil·lisegons $= 16$ segons.

(d) Encara que tots dos tenen un cost quadràtic amb la talla del problema, el cost de `Alg. B` creix més lentament que el de `Alg. A`, és a dir, és millor asimptòticament. Per això, en la práctica utilitzaria el `Alg. B`.