PRG - ETSInf. TEORIA. Curs 2013-14. Recuperació Parcial 1.
23 de juny de 2014. Durada: 1h 50m.

1. $\boxed{\text{3 punts}}$ Es desitja dissenyar un mètode recursiu que determine si cert array d'enters `v` conté els elements inicials d'una successió de Fibonacci. Per a això l'array es proporciona amb almenys 3 elements. Cal recordar que la successió de Fibonacci és la següent: 0,1,1,2,3,5,8 ... És a dir, el terme n-èsim és `n` si n = 0 o n = 1; en un altre cas, es calcula com la suma dels dos termes anteriors ((n-1)-èsim i (n-2)-èsim).

   (a) (2.5 punts). Escriure el mètode d'acord a les consideracions anteriors.

   Solució: mitjançant recorregut ascendent:

```
/** v.length >= 3, 2 <= i <= v.length */
public static boolean es_fibo (int[] v, int i) {
  if (i==v.length) return true;
  if (v[0]!=0 || v[1]!=1) return false;
  else return (v[i]==v[i-1]+v[i-2]) && es_fibo(v,i+1);
}
```

   mitjançant recorregut descendent:

```
/** v.length >= 3, 1 <= i <= v.length-1 */
public static boolean es_fibo (int[] v, int i) {
  if (i<=1) return (v[0]==0 && v[1]==1);
  else return (v[i]==v[i-1]+v[i-2]) && es_fibo(v,i-1);
}
```

   (b) (0.5 punts). D'acord a la implementació del mètode anterior, indicar la crida inicial perquè es verifique la propietat sobre tot l'array.

   Solució: Assumint `v` un array amb almenys 3 elements, la crida inicial, per a la solució recursiva ascendent, seria

   `es_fibo(v,2)`

   I per a la solució recursiva descendent:

   `es_fibo(v,v.length-1)`

2. $\boxed{\text{3 punts}}$ El següent mètode recursiu, `inversio(String)`, obté una `String` amb la inversió dels caràcters de la que rep com a argument. Per exemple, la inversió de la cadena `hola` és `aloh`.

```
public static String inversio (String s) {
    if (s.length() <= 1) return s;
    else return inversio(s.substring(1)) + s.charAt(0);
}
```

Es vol estudiar el seu cost temporal en les dues situacions següents:

1. Suposar que tant l'operació `substring(int)` com l'operació de concatenació (operador `+`) tenen un **cost constant** amb la llargària de la String `s`.
2. Suposar que l'operació `substring(int)` té un **cost lineal** amb la llargària de la String `s`, mentre que l'operació de concatenació (operador `+`) té un **cost constant** amb el nombre total de caràcters que es concatenen.

Per a cadascuna de les dues situacions **es demana**:

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

Solució: Per a ambdues situacions, la talla del problema és el nombre de caràcters de la `String s`, que canviarà en cada crida recursiva. L'expressió que la defineix és `s.length`, que anomenarem `n`.

(b) (0.5 punts) Indicar si existeixen diferents instàncies significatives per al cost temporal de l'algorisme, i identificar-les si és el cas.

Solució: Per a ambdues situacions, no existeixen instàncies significatives perquè es tracta d'un recorregut sobre la `String s`.

(c) (1.5 punts) Escriure les equacions de recurrència del cost temporal en funció de la talla, resolent-les per substitució.

Solució:
Per al cas de `substring` i operador de concatenació amb costos constants:

$$T(n) = \begin{cases} k' & si\ n \leq 1 \\ T(n-1) + k & si\ n > 1 \end{cases}$$

sent k i k' constants positives, en alguna unitat de temps. Resolent per substitució:

$$\begin{aligned} T(n) &= T(n-1) + k = T(n-2) + 2k = T(n-3) + 3k = \cdots = \\ &= T(n-i) + i \cdot k = \cdots = \\ &(cas\ base:\ n-i=1,\ i=n-1) \\ &= T(1) + (n-1)k = kn - k + k' \end{aligned}$$

Per al cas de `substring` amb cost lineal i operador de concatenació amb cost constant:

$$T(n) = \begin{cases} k' & si\ n \leq 1 \\ T(n-1) + kn & si\ n > 1 \end{cases}$$

sent k i k' constants positives, en alguna unitat de temps. Resolent per substitució i menyspreant termes d'ordre inferior:

$$\begin{aligned} T(n) &= T(n-1) + kn = T(n-2) + k(n-1) + kn = T(n-3) + k(n-2) + k(n-1) + kn = \cdots = \\ &= T(n-i) + \textstyle\sum_{j=n-(i-1)}^{n} kj = \cdots = \\ &(cas\ base:\ n-i=1,\ i=n-1,\ n-(i-1)=2) \\ &= T(1) + \textstyle\sum_{j=2}^{n} kj = k' + k\Big(\dfrac{n(n+1)}{2} - 1\Big) \end{aligned}$$

(d) (0.5 punts) Expressar el resultat anterior en notació asimptòtica.

Solució:
Per al cas de `substring` i operador de concatenació amb costos constants: $T(n) \in \theta(n)$.
Per al cas de `substring` amb cost lineal: $T(n) \in \theta(n^2)$.

(e) (0.25 punts) Quines de les dues situacions creus que és la més favorable des d'un punt de vista de cost temporal? Justifica la teva resposta.

Solució: A la vista del cost temporal asimptòtic és més favorable la situació on `substring` presenta cost constant ja que l'algorisme té un cost lineal amb la talla del problema.

3. $\boxed{\text{4 punts}}$ El següent algorisme iteratiu retorna un enter corresponent a la suma màxima dels valors emmagatzemats en posicions consecutives d'un array donat `a`.

```
/** Precondició: a.length >= 1 */
public static int metode (int[] a) {
    int n = a.length, max = a[0];
    for (int i=0; i<=n-1; i++) {
        int suma = 0;
        for (int j=i; j<=n-1; j++) {
            suma = suma + a[j];
            if ( suma > max ) max = suma;
        }
    }
    return max;
}
```

Es demana:

a) (0.5 punts) Indicar quina és la mida o talla del problema, així com l'expressió que la representa.

> **Solució:** La talla és la grandària de l'array, `a.length`, que anomenarem `n`.

b) (0.5 punts) Indicar si existeixen diferents instàncies significatives per al cost temporal de l'algorisme, i identificar-les si és el cas.

> **Solució:** No hi ha instàncies significatives, perquè és un problema de recorregut d'un array.

c) (2 punts) Escollir una unitat de mesura per al cost (passos de programa, instrucció crítica) i d'acord amb ella, obtenir una expressió el més precisa possible del cost temporal del programa (per al cas millor i el cas pitjor si és el cas).

> **Solució:**
> En passos de programa:
>
> $$T(n) = 1 + \sum_{i=0}^{n-1}(1 + \sum_{j=i}^{n-1} 1) = 1 + \sum_{i=0}^{n-1}(1 + n - i) = 1 + n + n^2 - \frac{n(n-1)}{2} = \frac{1}{2}n^2 + \frac{3}{2}n + 1 \ p.p.$$
>
> Si prenem com instrucció crítica: `suma = suma + a[j];` i considerant-la de cost unitari:
>
> $$T(n) = \sum_{i=0}^{n-1}\sum_{j=i}^{n-1} 1 = \sum_{i=0}^{n-1}(n-i) = n\sum_{i=0}^{n-1} 1 - \sum_{i=0}^{n-1} i = n^2 - \frac{n(n-1)}{2} = n^2 - \frac{1}{2}(n^2 - n) = \frac{1}{2}n^2 + \frac{1}{2}n \ \ \textit{ins. crítiques}$$

d) (1 punt) Expressar el resultat anterior en notació asimptòtica.

> **Solució:** El cost és quadràtic amb la talla del problema, $T(n) \in \theta(n^2)$.