

Primer Parcial de Pràctiques de PRG

ETSIInf - Curs 2012/13

22 d'abril de 2013. Duració: 1 hora

1. 2.5 punts Considera el següent algorisme, vist en pràctiques, per a resoldre el problema de les “Torres d'Hanoi”:

```
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 qüestions següents:

- Sabent que per a resoldre el problema en el cas d'una torre de 4 discos l'algorisme farà 15 moviments de disc, quants en farà si la torre conté 5 discos?
- Quina hauria de ser la crida inicial a l'algorisme anterior si es vullguera moure una torre de 12 discos des de l'agulla 'z' fins a la 'h' fent servir com a agulla auxiliar la 's'?

Solució:

- El nombre de moviments per a una torre de 5 discos és dues vegades el nombre de moviments per a una torre de 4 discos més un moviment addicional, açò és: 31 moviments.
- La crida inicial seria: `hanoi(12,'z','h','s');`

2. 2.5 punts Indica què s'ha de modificar en el codi següent per a que el mètode `subcadena(String,String)` siga correcte (sabent que el mètode `prefixe(String,String)` ja està fet i funciona correctament):

```
/** Torna true sii a és subcadena de b.
 * @param a. String.
 * @param b. L'altra String.
 * @return boolean: cert sii a és subcadena de b. */
public static boolean subcadena(String a, String b) {
    if (a.length()<=b.length())
        return prefixe(a,b) || subcadena(a,b.substring(1));
}
```

NOTA: recorda que la crida `s.substring(int)` torna una nova `String` amb els caràcters de `s` des de la posició que es rep com argument fins a la fi de `s`.

Solució:

Falta el cas base, de forma que si la condició inicial no es compleix hauria de tornar `false`. Això és, escrit correctament seria:

```

public static boolean subcadena(String a, String b) {
    if (a.length() <= b.length())
        return prefixe(a,b) || subcadena(a,b.substring(1));
    else return false;
}

```

3. 2.5 punts S'ha fet una aproximació amb l'ordre fit dels temps mesurats per a un algorisme, la talla del qual és el nombre d'elements d'un array, i els resultats obtinguts són els que es mostren a continuació:

```

gnuplot> f(x)=a*x+b
gnuplot> fit f(x) "algorisme.out" using 1:2 via a,b
...
Final set of parameters          Asymptotic Standard Error
=====
a              = 0.599844        +/- 0.005358      (0.8932%)
b              = 1.00006         +/- 3324          (3.324e+05%)

```

A partir d'aquests resultats, fes una predicció del temps d'execució (en nanosegons) requerit per a un array de 10^8 elements.

Solució:

Per un array de 10^8 elements, el temps requerit per l'execució seria de $10^8 \cdot 0.599844 + 1.00006 \approx 59984401$ nanosegons.

4. 2.5 punts S'ha mesurat el temps d'execució de l'algorisme d'inserció directa per al cas promedi mitjançant les següents instruccions:

```

public static void mesuraInsercio() {
    int[] a;          // Array del problema
    int t, r;         // Comptadors de talla i repeticions
    long tmed1, tmed2, tmedt; // Temps
    // Imprimir capçalera de resultats per a Inserció Directa
    System.out.printf("# INSERCIÓ DIRECTA \n");
    System.out.printf("# Talla   Promedi (en microsegons)\n");
    System.out.printf("#-----\n");
    // Repetir el procés per a distintes talles
    for(t=INITALLA; t<=MAXTALLA; t+=INCRTALLA) {
        // Crear i omplir l'array amb valors aleatoris
        a = new int[t];
        arrayAleatori(a);
        tmedt = 0; // Temps acumulat inicial a 0
        for(r=0; r<REPETICIONS; r++) {
            tmed1 = System.nanoTime(); // Temps inicial
            AlgorismesMesurables.insercio(a);
            tmed2 = System.nanoTime(); // Temps final
            tmedt+=(tmed2-tmed1);      // Actualitzar temps acumulat
        }
        tmedt = tmedt/REPETICIONS; // Temps promedi del cas promedi
    }
    // Imprimir resultats
    System.out.printf("%8d  %10.3f\n",t,tmedt/Math.pow(10,3));
}

```

```
}  
}
```

Obtenint-se la taula de temps següent:

```
# INSERCIÓ DIRECTA  
# Talla      Promedi (en microsegundos)  
#-----  
10000        44,247  
20000        80,929  
30000        123,136  
40000        166,019  
50000        209,868  
60000        254,777  
70000        300,706  
80000        347,435  
90000        395,428  
100000       443,917
```

A la vista d'aquestos resultats, són els esperats per al cas promedi? Hi ha algun error en el codi anterior? Com el corregiries?

Solució:

Els resultats de temps obtinguts mostren un creixement lineal quan hauria de ser quadràtic. L'error està en on es fa la inicialització de l'array `a` ja que, per a una talla `t` donada, després de la primera crida al mètode `insercio` l'array ja està ordenat ascendentment (cas millor de la inserció directa). La crida `arrayAleatori(a);` hauria de ser la primera instrucció del bucle `for(r=0; r<REPETICIONS; r++)` per tal que en cada repetició l'array s'inicialitzara amb valors aleatoris.