Recuperación parcial 1 - Prácticas - PRG - ETSInf - Curso 2012/13.
17 de junio de 2013. Duración: 50 minutos.

NOMBRE Y GRUPO DE PRÁCTICAS:

1. 2 puntos Considera el siguiente algoritmo visto en prácticas:

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

Contesta a las siguientes cuestiones:

(a) Sabiendo que en el caso de una torre de 10 discos el algoritmo tarda 10 milisegundos, ¿cuántos tardaría si la torre tuviese 11 discos?

(b) ¿y si tuviera 12?

Solución:

(a) El tiempo de `hanoi` es exponencial, $T(n) \in \Theta(2^n)$ y por lo tanto $T(n) = k \times 2^n$. El valor de la constante k se puede conocer a partir del dato del enunciado $T(10) = k \times 2^{10} = 10$, $k = 10 \times 2^{-10}$. Así, si el número de discos es 11 se tiene que $T(11) = k \times 2^{11} = 20$ milisegundos.

(b) Si el número de discos es 12 se tiene que $T(12) = k \times 2^{12} = 40$ milisegundos.

2. 3 puntos Implementar un método **recursivo** con el siguiente perfil:

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

que compruebe si la `String a` es sufijo de la `String b`. Una cadena de caracteres es sufijo de otra cadena si todos sus caracteres aparecen, en el mismo orden, en las últimas posiciones de la segunda cadena. Se considera que una cadena vacía es sufijo de cualquier cadena.

NOTA: En la resolución de este ejercicio **sólo** se pueden usar los siguientes métodos de la clase `String`:

- `s.charAt(i)` devuelve el carácter que ocupa la posición `i` en `s`.
- `s.substring(i,j)` devuelve una subcadena de `s` formada por los caracteres comprendidos entre las posiciones `i` y `j-1`.
- `s.length()` devuelve la longitud de `s`.

Solución:

```
public static boolean esSufijo(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) &&
                  esSufijo(a.substring(0,a.length()-1),
                           b.substring(0,b.length()-1));
}
```

3. $\boxed{\text{5 puntos}}$ Considera la tabla siguiente con los tiempos de ejecución de dos algoritmos expresados en milisegundos:

# 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 las siguientes cuestiones:

(a) ¿Cuál es el coste asintótico que se ajusta mejor a los datos de la columna `Alg. A`?

(b) ¿Cuál es el coste asintótico que se ajusta mejor a los datos de la columna `Alg. B`?

(c) Escribe una estimación del tiempo **en segundos** que utilizaría cada algoritmo para una talla de 150000.

(d) ¿Qué algoritmo tiene un comportamiento asintótico mejor? ¿Cuál utilizarías en la práctica?

Solución:

(a) El coste asintótico que se ajusta mejor a los datos de la columna `Alg. A` es un coste cuadrático. Si llamamos n a la talla del problema, $T(n) \in \Theta(n^2)$. Se puede comprobar que cuando la talla se duplica (de n a $2n$), el tiempo se multiplica por 4 (de n^2 a 2^2n^2). Por ejemplo, para $n = 20000$ el tiempo es 448.906 milisegundos y para $n = 40000$ el tiempo es 1797.765 milisegundos, aproximadamente 4 veces el de $n = 20000$.

(b) El coste asintótico que se ajusta mejor a los datos de la columna `Alg. B` también es un coste cuadrático. Por ejemplo, para $n = 20000$ el tiempo es 289.250 milisegundos y para $n = 40000$ el tiempo es 1160.216 milisegundos, aproximadamente 4 veces el de $n = 20000$.

(c) Para una talla de 150000 elementos, el tiempo de ejecución del `Alg. A` sería aproximadamente $10^2 \times 252.470 \approx 25000$ milisegundos $= 25$ segundos.

El tiempo de ejecución del `Alg. B`, para una talla de 150000 elementos, sería aproximadamente $10^2 \times 160.113 \approx 16000$ milisegundos $= 16$ segundos.

(d) Aunque ambos tienen un coste cuadrático con la talla del problema, el coste de `Alg. B` crece más lentamente que el de `Alg. A`, es decir, es mejor asintóticamente. Por ello, en la práctica utilizaría el `Alg. B`.