Algoritmos y Estructuras de Datos II - 1° cuatrimestre 2013

Práctico 1 - Parte 2

1. El siguiente programa calcula el mínimo elemento de un arreglo a : **array**$[1..n]$ **of int** mediante la técnica de programación *divide y vencerás*. Analizá la eficiencia de $minimo(1, n)$.

   fun $minimo(i, k : \textbf{int})$ **ret** $m : \textbf{int}$
       if $i = k$ **then** $m := a[i]$
       else
           $j := (i + k)$ **div** 2
           $m := min(minimo(i, j), minimo(j+1,k))$

2. Dado un arreglo a : **array**$[1..N]$ **of nat** se define una *cima* de a como un valor k en el intervalo $1, \ldots, N$ tal que $a[1..k]$ está ordenado crecientemente y $a[k..N]$ está ordenado decrecientemente.

   a) Escribí un programa que encuentre la cima de un arreglo dado (asumiendo que efectivamente existe una cima), utilizando la idea de *búsqueda binaria*.

   b) Calculá el orden de complejidad del programa.

3. Una secuencia de valores $x_1, ..., x_n$ se dice que tiene *orden cíclico* si existe un i con $1 \leq i \leq n$ tal que $x_i < x_{i+1} < \ldots < x_n < x_1 < \ldots < x_{i-1}$. Por ejemplo, la secuencia $5, 6, 7, 8, 9, 1, 2, 3, 4$ tiene orden cíclico (tomando $i = 6$).

   Utilizá una idea similar a la *búqueda binaria* para escribir un programa que dado un arreglo a : **array**$[1..n]$ que almacena una secuencia de valores que tiene orden cíclico, encuentre el menor elemento de la secuencia (es decir, el valor alojado en la posición i). El programa debe ser de $\mathcal{O}(log(n))$. Analizá la eficiencia en forma detallada.

4. Determiná cuáles de las siguientes afirmaciones son verdades y cuáles falsas. Justificá apropiadamente.

   a) $\mathcal{O}(f + g) = \mathcal{O}(max(f, g))$.

   b) Si $s \in \mathcal{O}(f)$ y $r \in \mathcal{O}(g)$ entonces $s + r \in \mathcal{O}(f + g)$.

   c) Si $s \in \mathcal{O}(f)$ y $r \in \mathcal{O}(g)$ entonces $s - r \in \mathcal{O}(f - g)$.

   d) $2^{n+1} \in \mathcal{O}(2^n)$.

   e) $(m + 1)! \in \mathcal{O}(m!)$.

5. Ordenar utilizando $\subset, \subseteq$ e $=$ los órdenes de las siguientes funciones. No calcular límites, utilizar las propiedades algebraicas.

   $n \log 2^n$ $2^n \log n$ $n! \log n$ 2^n

6. Determinar la relación $(\subset, \subseteq, =)$ entre los órdenes de las siguientes funciones:

   $n^4 + 2 \log n$ $\log(n^{n^4})$ $2^{4 \log n}$ 4^n $n^3 \log n$

   Justificar sin utilizar la regla del límite.

7. Resolvé las siguientes recurrencias:

   a) $t(n) = \begin{cases} 1 & \text{si } n = 1 \\ 4 & \text{si } n = 2 \\ 8t(n-1) - 15t(n-2) & \text{si } n > 2 \end{cases}$

   b) $t(n) = \begin{cases} 0 & \text{si } 1 \leq n \leq 2 \\ 1 & \text{si } n = 3 \\ 3t(n-2) - 2t(n-3) & \text{si } n > 3 \end{cases}$

   c) $t(n) = \begin{cases} 5 & \text{si } n = 1 \\ 3t(n-1) - 2^{n-1} & \text{si } n > 1 \end{cases}$

d) $t(n) = \begin{cases} 1 & \text{si } n = 1 \\ t(n-1) + n/2 & \text{si } n > 1 \end{cases}$

e) $t(n) = \begin{cases} 0 & \text{si } 1 \leq n \leq 2 \\ 3t(n-1) - 4t(n-3) & \text{si } n > 2 \end{cases}$

f) $t(n) = \begin{cases} 1 & \text{si } 0 \leq n \leq 1 \\ \frac{t(n-1)+t(n-2)+12n-16}{2} & \text{si } n > 2 \end{cases}$

8. Dada la siguiente recurrencia

$$t(n) = \begin{cases} 4 & \text{si } n = 1 \\ 2t(\lfloor n/2 \rfloor) + 2nlog(n) & \text{si } n > 1 \end{cases}$$

Demostrá que $t(n) \in \mathcal{O}(nlog^2(n))$.

9. Calculá el orden de complejidad de los siguientes programas:

a)
```
proc f1(in n : nat)
    if n ≤ 1 then skip
    else
        for i := 1 to 8 do f1(n div 2) od
        for i := 1 to n³ do operación_de_O(1) od
```

b)
```
proc f2(in n : nat)
    for i := 1 to n do
        for j := 1 to i do operación_de_O(3) od
    od
    if n ≤ 0 then skip
    else
        for i := 1 to 4 do f2(n div 2) od
```

c)
```
proc f3(n : nat)
    for j := 1 to 6 do
        if n ≤ 1 then skip
        else
            for i := 1 to 3 do f3(n div 4) od
            for i := 1 to n⁴ do operación_de_O(1) od
    od
```

10. Sean K y L constantes, y f el siguiente procedimiento:

```
proc f(in n : nat)
    if n ≤ 1 then skip
    else
        for i := 1 to K do f(n div L) od
        for i := 1 to n⁴ do operación_de_O(1) od
```

Determiná posibles valores de K y L de manera que el procedimiento tenga orden:

a) $\Theta(n^4 \log n)$ b) $\Theta(n^4)$ c) $\Theta(n^5)$