Prueba de Análisis de Algoritmos – Curso 2015-2016 – 17 de marzo de 2016

Diseño y Análisis de Algoritmos – Grado en Ingeniería Informática

Valor: 30 % de la nota final. Duración: **1 hora y 15 minutos**

Se compensa el examen con 4 puntos de los 10 posibles

Soluciones

Ejercicio 1 [1.5 puntos]

Indicad la definición matemática de $\mathcal{O}$, y usadla para demostrar (sin usar límites), si se verifica:

$$5n \in \mathcal{O}(n\log(n))$$

Solución:

La definición de $\mathcal{O}$ es:

$$\mathcal{O}(g(n)) = \Big\{f(n) : \exists\ c > 0 \text{ y } n_0 > 0 \ / \ 0 \leq f(n) \leq c \cdot g(n), \forall n \geq n_0\Big\}$$

Por tanto, tenemos que ver si es posible encontrar una pareja de constantes $c > 0$ y $n_0 > 0$ de forma que se cumpla la definición. Empezamos analizando:

$$5n \leq cn\log(n)$$

Escogemos un valor de c lo suficientemente grande. En este caso $c = 5$ resulta adecuado para continuar:

$$5n \leq 5n\log(n)$$

$$0 \leq 5n\log(n) - 5n$$

$$0 \leq 5n(\log(n) - 1)$$

En este caso, sea cual sea la base del logaritmo, por ejemplo b, siempre es posible encontrar un n_0 de tal manera que se cumpla la definición. Bastaría escoger $n_0 = b$, ya que $(\log_b(n) - 1) \geq 0$ para todo $n \geq b$. Por tanto, podemos afirmar que $5n \in \mathcal{O}(n\log(n))$.

Ejercicio 2 [2.5 puntos]

La fórmula para considerar todas las operaciones que se llevan a cabo en un bucle de tipo FOR o WHILE es:

$$T_{\text{bucle}} = 1_{\text{inicialización}} + \sum^{n} (1_{\text{comparación}} + T_{\text{cuerpo}} + 1_{\text{incremento}}) + 1_{\text{última comparación}}$$

Utilízala para hallar el número de operaciones ($T(n)$, simplificada) del siguiente código:

```
int i=0;
while (i<n){
    int j=i;
    while (j<=n){
        procesa(i,j);   // una operación
        j++;
    }
    i++;
}
```

Se considera que las inicializaciones, comparaciones, e incrementos siempre necesitan una sola operación.

Solución:

El código consta de 2 bucles, que podemos descomponer de la siguiente manera:

$$T(n) = 1 + \sum_{i=0}^{n-1} (1 + T_{\text{While interno}} + 1) + 1$$

$$T_{\text{While interno}} = 1 + \sum_{j=i}^{n} (1 + 1 + 1) + 1 = 2 + 3(n - i + 1)$$

Sustituyendo en $T(n)$ tenemos:

$$T(n) = 2 + \sum_{i=0}^{n-1} \big(2 + 2 + 3(n - i + 1)\big) = 2 + \sum_{i=0}^{n-1} (7 + 3n - 3i)$$

$$= 2 + 7n + 3n^2 - 3 \sum_{i=0}^{n-1} i = 2 + 7n + 3n^2 - 3\frac{n(n-1)}{2}$$

Finalmente:

$$T(n) = 2 + 7n + 3n^2 - \frac{3}{2}n^2 + \frac{3}{2}n = \frac{3}{2}n^2 + \frac{17}{2}n + 2$$

Ejercicio 3 [3 puntos]

Halla una expresión no recursiva para la siguiente recurrencia por el método de **expansión de recurrencias**:

$$T(n) = 4T(n/2) + n$$

donde n es una potencia de 2 ($n = 2^k$, para $k = 1, 2, \ldots$), y donde $T(1) = 1$. Indica además el orden de $T(n)$.

Solución:

Procedemos a expandir la recurrencia:

$$\begin{aligned}
T(n) &= 4T(n/2) + n \\
&= 4\left[4T(n/2^2) + \frac{n}{2}\right] + n = 4^2T(n/2^2) + 2n + n \\
&= 4^2\left[4T(n/2^3) + \frac{n}{2^2}\right] + 2n + n = 4^3T(n/2^3) + 4n + 2n + n \\
&= 4^3\left[4T(n/2^4) + \frac{n}{2^3}\right] + 4n + 2n + n = 4^4T(n/2^4) + 8n + 4n + 2n + n \\
&\vdots \\
&= 4^iT(n/2^i) + n\sum_{j=0}^{i-1} 2^j = 4^iT(n/2^i) + (2^i - 1)n
\end{aligned}$$

Se llega al caso base cuando $n/2^i = 1$. Es decir, cuando $n = 2^i$. En ese caso, $n^2 = (2^i)^2 = (2^2)^i = 4^i$. Sustituyendo:

$$T(n) = n^2 + n^2 - n = 2n^2 - n$$

Ejercicio 4 [3 puntos]

Halla una expresión no recursiva para la siguiente recurrencia por el método **general de resolución recurrencias**:

$$T(n) = T(n/2) + \log_2 n$$

donde n es una potencia de 2 ($n = 2^k$, para $k = 1, 2, \ldots$), y donde $T(1) = 1$. Indica además el orden de $T(n)$.

Solución:
Para poder aplicar el método primero tenemos que hacer el cambio de variable $n = 2^k$ (donde $k = \log_2 n$):

$$T(n) = T(2^k) = T(2^k/2) + \log_2 2^k = T(2^{k-1}) + k$$

A continuación hacemos un cambio de función $t(k) = T(2^k)$:

$$T(n) = T(2^k) = t(k) = t(k-1) + k$$

El polinomio característico de la recurrencia es:

$$(x-1)(x-1)^2 = (x-1)^3$$

Por tanto, la recurrencia tiene la siguiente forma:

$$t(k) = C_1 1^k + C_2 k 1^k + C_3 k^2 1^k = C_1 + C_2 k + C_3 k^2$$

Para hallar las constantes necesitamos dos casos base adicionales. Procedemos a calcular $T(2) = T(1) + \log_2 2 = 2$, y $T(4) = T(2) + \log_2 4 = 4$. Los correspondientes casos base para $t(k)$ son:

$$\begin{aligned} T(1) = T(2^0) = t(0) &= 1 \\ T(2) = T(2^1) = t(1) &= 2 \\ T(4) = T(2^2) = t(2) &= 4 \end{aligned}$$

Por tanto, el sistema de ecuaciones a resolver es:

$$\left.\begin{array}{rcrcl} t(0) & = & C_1 & = & 1 \\ t(1) & = & C_1 + C_2 + C_3 & = & 2 \\ t(2) & = & C_1 + 2C_2 + 4C_3 & = & 4 \end{array}\right\}$$

Las soluciones son $C_1 = 1$ y $C_2 = C_3 = 1/2$. Por tanto, $t(k) = 1 + k/2 + k^2/2$. Deshaciendo el cambio de variable, donde $k = \log_2 n$, obtenemos:

$$t(k) = T(2^k) = T(n) = 1 + \frac{1}{2}\log_2 n + \frac{1}{2}(\log_2 n)^2 \in \theta\big((\log n)^2\big)$$