Universidad Rey Juan Carlos

Prueba de Análisis de Algoritmos – Curso 2014-2015 – 15 de junio de 2015

Diseño y Análisis de Algoritmos – Grado en Ingeniería Informática

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

Se compensa el examen con 4 puntos de los 10 posibles

Ejercicio 1 [1 punto]

Determinar el orden de complejidad (no es necesario hallar las constantes) de la siguiente función recursiva:

$$T(n) = 5T(n-1) - 8T(n-2) + 4T(n-3) + 3 + n + 2^n + n^3 2^n$$

Solución:

Aplicamos el método general de resolución de recurrencias. Primero pasamos los términos con T del lado derecho al izquierdo, y expresamos el resto de términos de la derecha como polinomios por potencias elevadas a n:

$$T(n) - 5T(n-1) + 8T(n-2) - 4T(n-3) = (3+n)1^n + (1+n^3)2^n$$

El polinomio característico es:

$$(x^3 - 5x^2 + 8x - 4)(x-1)^2(x-2)^4$$

Factorizamos el primer término aplicando el método de Ruffini y agrupamos términos:

$$(x-1)(x-2)^2(x-1)^2(x-2)^4 = (x-1)^3(x-2)^6$$

Por tanto, la función $T(n)$ tiene la siguiente expresión:

$$T(n) = C_1 + C_2 n + C_3 n^2 + C_4 2^n + C_5 n 2^n + C_6 n^2 2^n + C_7 n^3 2^n + C_8 n^4 2^n + C_9 n^5 2^n$$

Finalmente, el orden de complejidad es:

$$T(n) \in \theta(n^5 2^n)$$

Ejercicio 2 [2 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:

```
for (int i=0; i<n; i++){
    int j=i;
    while (j<=n){
        procesa(i,j);   // dos operaciones

        j++;
    }
}
```

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:

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

$$T_{\text{For 1}} = 1 + \sum_{j=i}^{n} (1 + 2 + 1) + 1 = 2 + 4(n - i + 1)$$

Sustituyendo en $T(n)$ tenemos:

$$T(n) = 2 + \sum_{i=0}^{n-1} \big(2 + 2 + 4(n - i + 1)\big) = 2 + \sum_{i=0}^{n-1} (8 + 4n - 4i)$$

$$= 2 + 8n + 4n^2 - 4\sum_{i=0}^{n-1} i = 2 + 8n + 4n^2 - 4\frac{n(n-1)}{2}$$

Finalmente:

$$T(n) = 2 + 8n + 4n^2 - 2n^2 + 2n = 2n^2 + 10n + 2$$

Ejercicio 3 [3.5 puntos]

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

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

donde n es una potencia de 4 ($n = 4^k$, para $k = 1, 2, \ldots$), y donde $T(1) = 0$. Indica además el orden de $T(n)$.

Solución:

Procedemos a expandir la recurrencia:

$$\begin{aligned}
T(n) &= 2T(n/4) + n \\
&= 2\left[2T(n/4^2) + \frac{n}{4}\right] + n = 4T(n/4^2) + \frac{n}{2} + n \\
&= 4\left[2T(n/4^3) + \frac{n}{4^2}\right] + \frac{n}{2} + n = 8T(n/4^3) + \frac{n}{4} + \frac{n}{2} + n \\
&= 8\left[2T(n/4^4) + \frac{n}{4^3}\right] + \frac{n}{4} + \frac{n}{2} + n = 16T(n/4^4) + \frac{n}{8} + \frac{n}{4} + \frac{n}{2} + n \\
&\vdots \\
&= 2^i T(n/4^i) + n\sum_{j=0}^{i-1} \frac{1}{2^j} = 2^i T(n/4^i) + n\frac{2 \cdot 2^i - 2}{2^i}
\end{aligned}$$

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

$$T(n) = \sqrt{n} \cdot 0 + n\frac{2\sqrt{n} - 2}{\sqrt{n}} = \sqrt{n}(2\sqrt{n} - 2) = 2n - 2\sqrt{n}$$

Ejercicio 4 [3.5 puntos]

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

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

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 $k = \log_2 n$):

$$T(n) = T(2^k) = 2T(2^k/2) + k2^k = 2T(2^{k-1}) + k2^k$$

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

$$T(n) = T(2^k) = t(k) = 2t(k-1) + k2^k$$

El polinomio característico de la recurrencia es:

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

Por tanto, la recurrencia tiene la siguiente forma:

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

Para hallar las constantes necesitamos dos casos base adicionales. Procedemos a calcular $T(2) = 2T(1) + 2 = 4$, y $T(4) = 2T(2) + 4 \cdot 2 = 16$. Los correspondientes casos base para $t(k)$ son:

$$\begin{array}{rcl} T(1) = T(2^0) = t(0) & = & 1 \\ T(2) = T(2^1) = t(1) & = & 4 \\ T(4) = T(2^2) = t(2) & = & 16 \end{array}$$

Por tanto, el sistema de ecuaciones a resolver es:

$$\left.\begin{array}{rcrcr} t(0) & = & C_1 & = & 1 \\ t(1) & = & 2^1C_1 + 2^1C_2 + 2^1C_3 & = & 4 \\ t(2) & = & 2^2C_1 + 2 \cdot 2^2C_2 + 4 \cdot 2^2C_3 & = & 16 \end{array}\right\}$$

Las soluciones son $C_1 = 1$ y $C_2 = C_3 = 1/2$. Por tanto, $t(k) = 2^k + k2^k/2 + k^2 2^k/2$. Deshaciendo el cambio de variable, donde $k = \log_2 n$, obtenemos:

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