Universidad Rey Juan Carlos **B**

Prueba de Análisis de Algoritmos – Curso 2014-2015 – 10 de abril de 2015

Diseño y Análisis de Algoritmos – Grado en Ingeniería Informática

Valor: 30 % de la nota final. Duración: **1 hora y 45 minutos**

Se compensa el examen con 12 puntos de los 30 posibles

Ejercicio 1 [3 puntos]

Demostrar **mediante la definición** de complejidad asintótica Ω, sin usar límites, si se verifica:

$$n^2 - 3n + 1 \in \Omega(n^2)$$

Solución:

La definición de Ω es:

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

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:

$$n^2 - 3n + 1 \geq cn^2$$

Escogemos un valor de c lo suficientemente pequeño, pero positivo. En este caso $c = 1/2$ resulta ser suficiente. Por tanto, tenemos:

$$n^2 - 3n + 1 \geq \frac{1}{2}n^2$$

$$\frac{1}{2}n^2 - 3n + 1 \geq 0$$

$$n^2 - 6n + 2 \geq 0$$

La función de la izquierda es una parábola convexa, así que ya podemos ver que va a ser posible encontrar un valor para n_0. Para ello, analizamos la mayor raíz del polinomio, que es:

$$\frac{6 + \sqrt{36 - 8}}{2} = \frac{6 + \sqrt{28}}{2} = 3 + \sqrt{7}$$

Por tanto, podemos escoger para n_0 cualquier valor superior o igual a $3 + \sqrt{7}$. Por ejemplo, $n_0 = 6$, ya que $\sqrt{7} < 3$. De esta manera hemos encontrado las dos constantes c y n_0 que satisfacen la definición, y podemos afirmar que $n^2 - 3n + 1 \in \Omega(n^2)$.

Ejercicio 2 **[4 puntos]**

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

$$T(n) = 4T(n-1) - 5T(n-2) + 2T(n-3) + 1 + n^2 + n2^n$$

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:

$$T(n) - 4T(n-1) + 5T(n-2) - 2T(n-3) = (1 + n^2)1^n + n2^n$$

El polinomio característico es:

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

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

$$(x-1)^2(x-2)(x-1)^3(x-2)^2 = (x-1)^5(x-2)^3$$

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

$$T(n) = C_1 + C_2 n + C_3 n^2 + C_4 n^3 + C_5 n^4 + C_6 2^n + C_7 n2^n + C_8 n^2 2^n$$

Finalmente, el orden de complejidad es:

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

Ejercicio 3 [6 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++)
     for(j=i;j<=n;j++){
          int k=0;
          while (k<j){
               procesa(i,j,k);   // una operación
               k++;
          }
     }
```

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

Solución:

El código consta de 3 bucles, que podemos descomponer de la siguiente manera:

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

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

$$T_{\text{For 2}} = 1 + \sum_{k=0}^{j-1}(1 + 1 + 1) + 1 = 2 + 3j$$

Sustituyendo en $T_{\text{For 1}}$ tenemos:

$$T_{\text{For 1}} = 2 + \sum_{j=i}^{n}(4 + 3j)$$

$$= 2 + \sum_{j=i}^{n} 4 + 3\sum_{j=i}^{n} j = 2 + \sum_{j=i}^{n} 4 + 3\left[\sum_{j=1}^{n} j - \sum_{j=1}^{i-1} j\right]$$

$$= 2 + 4(n - i + 1) + 3\left[\frac{n(n+1)}{2} - \frac{(i-1)i}{2}\right]$$

Simplificando obtenemos:

$$T_{\text{For 1}} = \frac{3n^2 + 11n + 12 - 3i^2 - 5i}{2}$$

Sustituyendo en la expresión para el primer bucle tenemos:

$$T(n) = 2 + \sum_{i=0}^{n-1} \left(2 + \frac{3n^2 + 11n + 12 - 3i^2 - 5i}{2} \right)$$

$$T(n) = 2 + \frac{1}{2} \sum_{i=0}^{n-1} \left(3n^2 + 11n + 16 - 3i^2 - 5i \right)$$

$$T(n) = 2 + \frac{3}{2}n^3 + \frac{11}{2}n^2 + 8n - \frac{3}{2} \sum_{i=0}^{n-1} i^2 - \frac{5}{2} \sum_{i=0}^{n-1} i$$

Para terminar el problema hay que resolver los dos últimos sumatorios:

$$T(n) = 2 + \frac{3}{2}n^3 + \frac{11}{2}n^2 + 8n - \frac{3}{2}\frac{(2(n-1)+1)(n-1)n}{6} - \frac{5}{2}\frac{(n-1)n}{2}$$

$$T(n) = 2 + \frac{3}{2}n^3 + \frac{11}{2}n^2 + 8n - \frac{2n^3 - 3n^2 + n}{4} - \frac{5n^2 - 5n}{4}$$

Finalmente:

$$T(n) = \frac{4n^3 + 20n^2 + 36n + 8}{4} = n^3 + 5n^2 + 9n + 2 \in \theta(n^3)$$

Ejercicio 4 [6 puntos]

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

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

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:

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

Se llega al caso base cuando $n/2^i = 1$. Es decir, cuando $n = 2^i$, e $i = \log_2 n$. Sustituyendo:

$$T(n) = nT(1) + n\log_2 n = n + n\log_2 n \in \theta(n \log n)$$

Ejercicio 5 [6 puntos]

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

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

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:

Para poder aplicar el método primero tenemos que hacer el cambio de variable $n = 2^k$:

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

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

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

El polinomio característico de la recurrencia es:

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

Por tanto, la recurrencia tiene la siguiente forma:

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

Para hallar las constantes necesitamos un segundo caso base. Procedemos a calcular $T(2) = 2T(1) + 2 = 4$. Los correspondientes casos base para $t(k)$ son:

$$T(1) = T(2^0) = t(0) = 1$$
$$T(2) = T(2^1) = t(1) = 4$$

Por tanto, el sistema de ecuaciones a resolver es:

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

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

$$t(k) = T(2^k) = T(n) = n + n\log_2 n \in \theta(n \log n)$$

Ejercicio 6 [5 puntos]

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

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

donde $T(0) = 1$.

Solución:
En primer lugar podemos reescribir la fórmula de la siguiente manera:

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

A partir de ahora se puede resolver de varias formas.

- Primera forma (difícil). Expandimos $T(n-1)$, quedando:

$$T(n) = 1 + 1 + \sum_{i=0}^{n-2} T(i) + \sum_{i=0}^{n-2} T(i) = 2 + 2\left[\sum_{i=0}^{n-2} T(i)\right]$$

  Ahora extraemos $T(n-2)$ del sumatorio, y lo expandimos:

$$T(n) = 2 + 2\left[T(n-2) + \sum_{i=0}^{n-3} T(i)\right] = 2 + 2\left[1 + \sum_{i=0}^{n-3} T(i) + \sum_{i=0}^{n-3} T(i)\right]$$
$$= 2 + 2\left[1 + 2\sum_{i=0}^{n-3} T(i)\right] = 4 + 4\left[\sum_{i=0}^{n-3} T(i)\right]$$

  Hacemos lo mismo un paso más:

$$T(n) = 4+4\left[T(n-3) + \sum_{i=0}^{n-4} T(i)\right] = 4+4\left[1 + 2\sum_{i=0}^{n-4} T(i)\right] = 8+8\left[\sum_{i=0}^{n-4} T(i)\right]$$

  Tras una serie de pasos (en este caso usamos la variable j ya que la i ya aparece en el sumatorio) tenemos:

$$T(n) = 2^{j-1} + 2^{j-1}\left[\sum_{i=0}^{n-j} T(i)\right]$$

  Alcanzamos el caso base cuando el sumatorio solo suma un término $T(0)$. Esto sucede cuando $n = j$. Sustituyendo:

$$T(n) = 2^{n-1} + 2^{n-1}T(0) = 2 \cdot 2^{n-1} = 2^n \in \theta(2^n)$$

- Segunda forma (fácil). Por la definición de T tenemos:

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

Por tanto, sustituyendo en (1) tenemos una nueva expresión simplificada de la recurrencia:

$$T(n) = 2T(n-1)$$

Ahora procederíamos a resolverla:

$$T(n) = 2T(n-1) = 4T(n-2) = 8T(n-3) = \cdots = 2^i T(n-i)$$

El caso base se alcanza cuando $n-i=0$, es decir, cuando $n=i$. Sustituyendo:

$$T(n) = 2^n T(0) = 2^n \in \theta(2^n)$$