2.3 Ecuaciones de recurrencia

En el análisis de algoritmos se suele llegar en el cálculo del tiempo de ejecución a ecuaciones de recurrencia que se presentan en general como:

$$t(n) = \begin{cases} b & si\ n \leq n_0 \\ g(t(n), t(n-1), \ldots, t(n-k), n) & si\ n > n_0 \end{cases} \tag{2.3}$$

Ejemplo 2.4 En el algoritmo de las Torres de Hanoi tenemos:

```
Hanoi(n, i, j, k) /*Paso de n aros de i a j teniendo a k de pivote*/
    si n > 0
        Hanoi(n − 1, i, k, j)
        mover(i, j)
        Hanoi(n − 1, k, j, i)
    finsi
fin
```

donde:

$$t(n) = \begin{cases} 1 & si\ n = 0 \\ 2t(n-1) + 1 & si\ n > 0 \end{cases}$$

lo que da lugar a una ecuación de recurrencia.

2.3.1 Ecuaciones lineales homogéneas

El caso más sencillo será el de ecuaciones lineales homogéneas de coeficientes constantes:

$$a_0 t(n) + a_1 t(n-1) + \ldots + a_k t(n-k) = 0 \tag{2.4}$$

donde los coeficientes a_i son constantes. Se llama lineal porque la ecuación es lineal y homogénea porque no tiene término independiente de t.

Buscamos una solución de la forma $t(n) = x^n$ con x constante. De este modo:

$$a_0 x^n + a_1 x^{n-1} + \ldots + a_k x^{n-k} = 0$$

con lo que habrá que resolver la **ecuación característica**:

$$a_0 x^k + a_1 x^{k-1} + \ldots + a_k = 0 \tag{2.5}$$

Puede ocurrir que las **soluciones** de la ecuación característica sean todas **distintas** $(s_1, s_2, \ldots, s_k)$, en cuyo caso toda combinación lineal $t(n) = \sum_{i=1}^{k}(c_i s_i^n)$ es una solución de la recurrencia, y para determinar los valores de las constantes habrá que resolver el sistema formado imponiendo las condiciones iniciales.

Ejemplo 2.5 Como ejemplo veamos cuál es la solución de la recurrencia $t(n)-3t(n-1)-4t(n-2)=0$ cuando $n>2$, con condiciones iniciales $t(0)=0$ y $t(1)=1$.

La ecuación característica es $x^2-3x-4=0$, que tiene soluciones -1 y 4, por lo que $t(n)=c_1(-1)^n+c_2 4^n$, e imponiendo las condiciones iniciales tenemos:

$$\begin{array}{l} c_1+c_2=0 \\ -c_1+4c_2=1 \end{array}$$

de donde $c_1=-\frac{1}{5}$ y $c_2=\frac{1}{5}$, con lo que la solución es $t(n)=\frac{1}{5}(4^n-(-1)^n)$.

Puede ocurrir que las **soluciones** de la ecuación característica **no sean simples**.

Si s es una raíz de multiplicidad m, el polinomio característico se descompone como $p(x)=(x-s)^m q(x)$, con lo que todas las derivadas del polinomio $p(x)$ hasta la de grado $m-1$ se anulan para $x=s$.

De este modo, sea $x^{n-k}p(x)=0$ la ecuación polinómica que representa la ecuación de recurrencia original, su derivada es $(n-k)x^{n-k-1}p(x)+x^{n-k}p'(x)$, que se anula para $x=s$.

Por otro lado, la derivada es $a_0nx^{n-1}+a_1(n-1)x^{n-2}+\ldots+a_k(n-k)x^{n-k-1}$, y con $x=s$ queda $a_0ns^{n-1}+a_1(n-1)s^{n-2}+\ldots+a_k(n-k)s^{n-k-1}$, y multiplicando por s tendremos $a_0ns^n+a_1(n-1)s^{n-1}+\ldots+a_k(n-k)s^{n-k}$, con lo que se ve que $t(n)=ns^n$ es solución de la ecuación de recurrencia.

Del mismo modo se puede comprobar que siendo s una solución de multiplicidad m, $s^n, ns^n, n^2s^n, \ldots, n^{m-1}s^n$ son soluciones de la ecuación de recurrencia.

Ejemplo 2.6 Como ejemplo veamos cuál es la solución de la recurrencia $t(n)=5t(n-1)-8t(n-2)+4t(n-3)$ cuando $n>3$, con condiciones iniciales $t(0)=0$, $t(1)=1$ y $t(2)=2$.

La ecuación característica es $x^3-5x^2+8x-4=0$, que se descompone como $(x-1)(x-2)^2=0$, por lo que la solución general de la recurrencia es $t(n)=c_1 1^n+c_2 2^n+c_3 n 2^n$, e imponiendo las condiciones iniciales tenemos el sistema de ecuaciones:

$$\begin{array}{l} c_1+c_2=0 \\ c_1+2c_2+2c_3=1 \\ c_1+4c_2+8c_3=2 \end{array}$$

y resolviendo el sistema tenemos $t(n)=-2+2^{n+1}-n2^{n-1}$.

Este ejemplo de resolución de ecuación de recurrencia no puede corresponder a un caso real de cálculo de tiempo de ejecución pues el término de mayor orden en $t(n)$ aparece con coeficiente negativo, lo que corresponde a un tiempo de ejecución negativo cuando el tamaño de la entrada aumenta.

2.3.2 Caso general

Generalizando al caso:

$$a_0t(n) + a_1t(n-1) + \ldots + a_kt(n-k) = b^n p(n) \tag{2.6}$$

donde b es una constante y $p(n)$ un polinomio de grado d, la ecuación característica es:

$$\left(a_0x^k + a_1x^{k-1} + \ldots + a_k\right)(x-b)^{d+1} = 0 \tag{2.7}$$

Ejemplo 2.7 Este tipo de ecuaciones también se pueden resolver reduciendo el orden del polinomio $p(n)$.

Si tenemos

$$t(n) - 2t(n-1) = 3^n(n+1) \tag{2.8}$$

multiplicando por 3 tendremos:

$$3t(n) - 6t(n-1) = 3^{n+1}(n+1) \tag{2.9}$$

y sustituyendo en la ecuación 2.8 n por $n+1$ tenemos:

$$t(n+1) - 2t(n) = 3^{n+1}(n+2) \tag{2.10}$$

y restando a 2.10 la ecuación 2.9:

$$t(n+1) - 5t(n) + 6t(n-1) = 3^{n+1} \tag{2.11}$$

con lo que se ha disminuido en uno el grado de $p(n)$. Haciendo lo mismo repetidamente se llega, en un número finito de pasos, a una ecuación homogénea que se resuelve como se vio en la subsección anterior.

Para ecuaciones de la forma:

$$a_0t(n) + a_1t(n-1) + \ldots + a_kt(n-k) = b_1^n p_1(n) + b_2^n p_2(n) + \ldots \tag{2.12}$$

se tiene como ecuación característica:

$$\left(a_0x^k + a_1x^{k-1} + \ldots + a_k\right)(x-b_1)^{d_1+1}(x-b_2)^{d_2+1}\ldots = 0 \tag{2.13}$$

2.3.3 Técnica de la inducción constructiva

Para resolver algunas ecuaciones de recurrencia se puede utilizar la técnica de inducción, de modo que si suponemos que $t \in \theta(f)$ ($O(f)$ ó $\Omega(f)$) lo podemos demostrar por inducción.

Ejemplo 2.8 Como ejemplo vemos la ecuación de recurrencia:

$$t(n) = \begin{cases} a & si\ n = 1 \\ bn^2 + nt(n-1) & si\ n > 1 \end{cases}$$

Podemos suponer (pues es razonable hacerlo así) que $t(n) \in \theta(n!)$.

Para demostrar esto demostraremos por inducción que $t(n) \in O(n!)$ y que $t(n) \in \Omega(n!)$.

Empezamos demostrando que $t(n) \in \Omega(n!)$:

Está claro que $t(1) \geq v1!$ para un cierto valor constante de v que puede ser a.

Suponemos que $t(n-1) \geq v(n-1)!$. Como $t(n) = bn^2 + nt(n-1) \geq bn^2 + nv(n-1)! \geq vn!$, luego $t(n) \in \Omega(n!)$.

Demostrando ahora que $t(n) \in O(n!)$:

Está claro que $t(1) \leq u1!$ para ciertos valores constantes de u.

Suponemos que $t(n-1) \leq u(n-1)!$. Tenemos $t(n) = bn^2 + nt(n-1) \leq bn^2 + nu(n-1)!$, de donde no podemos deducir que $t(n) \in O(n!)$.

Suponemos que existen enteros positivos u y w tal que $t(n) \leq un! - wn$. Esto se cumplirá para $n = 1$ dependiendo de los valores de u y w.

Suponemos que se cumple para $t(n-1)$, por lo que $t(n) \leq bn^2 + n\,(u(n-1)! - w(n-1)) = un! + ((b-w)n + w)\,n$, y para que esta expresión sea menor o igual que $un! - wn$ tiene que ser $(b-w)n + w \leq -w$, de donde tiene que ser $w \geq b\frac{n}{n-2}$. Tomando $w = 2b$ se satisface la desigualdad y se cumple la recursión, y tomando $a = u - 2b$ se cumplirá también el caso base.

Lo difícil con este método puede ser acertar con la función.

2.3.4 Recurrencias asintóticas condicionales

En algunos casos puede interesar, para resolver una ecuación de recurrencia, restringirnos a valores de n que cumplan una cierta condición y posteriormente quitar la restricción utilizando el siguiente teorema:

Teorema 2.1 *Si $b \geq 2$ es un entero y $f : N \longrightarrow R^+$ una función que es no decreciente a partir de un valor n_0 (**eventualmente no decreciente**) y $f(bn) \in O(f(n))$ (f es **b-armónica**), y $t : N \longrightarrow R^+$ es eventualmente no decreciente tal que $t(n) \in \theta(f|n$ potencia de $b)$, entonces $t(n) \in \theta(f)$.*

Ejemplo 2.9 Si

$$t(n) = \begin{cases} a & si\ n = 1 \\ t\left(\lfloor \frac{n}{2} \rfloor\right) + t\left(\lceil \frac{n}{2} \rceil\right) + bn & con\ b\ positivo,\ si\ n > 1 \end{cases}$$

donde $\lfloor\ \rfloor$ representa al entero más próximo menor y $\lceil\ \rceil$ al entero más próximo mayor.

Será más fácil resolverlo suponiendo que n es una potencia de 2, pues en este caso tenemos:

$$t(n) = \begin{cases} a & si\ n = 1 \\ 2t\left(\frac{n}{2}\right) + bn & con\ b\ positivo,\ si\ n > 1,\ y\ n = 2^k \end{cases}$$

Podemos calcular su orden del siguiente modo:
si $n = 2^k$ tenemos $t\left(2^k\right) = 2t\left(2^{k-1}\right) + b2^k = 2\left(2t\left(2^{k-2}\right) + b2^{k-1}\right) + b2^k = 2^2 t\left(2^{k-2}\right) + 2b2^k = \ldots = an + bn\log n$ que es de orden $\theta(n\log n | n\ es\ potencia\ de\ 2)$.

A partir de esto, y utilizando el teorema 2.1, se puede calcular el orden exacto del tiempo de ejecución en el caso general: sólo hay que demostrar que t y $f(n) = n\log n$ son no decrecientes y que f es 2-armónica.

f es 2-armónica pues $f(2n) = 2n\log 2n = 2n\log n + 2n \in \theta(n\log n)$. Y además es eventualmente no decreciente pues $n\log n$ es creciente en todo su dominio.

Para demostrar que t es eventualmente no decreciente tenemos el problema de que sólo conocemos t por su ecuación de recurrencia. Se puede demostrar por inducción:

$t(2) = 2t(1) + b > t(1)$,
y suponiendo que es eventualmente no decreciente para valores menores o iguales a n demostraremos que $t(n+1) \geq t(n)$:
si n es par tenemos $\lfloor \frac{n}{2} \rfloor = \lfloor \frac{n+1}{2} \rfloor$ y $\lceil \frac{n}{2} \rceil + 1 = \lceil \frac{n+1}{2} \rceil$, por lo que $t(n+1) \geq t(n)$. Y similarmente se demuestra la desigualdad para n impar.

2.3.5 Cambio de variable

Algunas veces, para resolver ecuaciones de recurrencia, se puede hacer un cambio de variable para transformar la ecuación a uno de los tipos vistos en las subsecciones anteriores.

Ejemplo 2.10 Si

$$t(n) = \begin{cases} a & si\ n = 1 \\ 2t\left(\frac{n}{2}\right) + bn & con\ b\ positivo,\ si\ n > 1,\ y\ n = 2^k \end{cases}$$

haciendo el cambio $n = 2^k$ obtenemos:

$$t\left(2^k\right) = \begin{cases} a & si\ k = 0 \\ 2t\left(2^{k-1}\right) + b2^k & si\ k > 0 \end{cases}$$

La ecuación característica es $(x-2)^2$, por lo que la solución general es de la forma $t_k = c_1 2^k + c_2 k 2^k$ y, deshaciendo el cambio, es $t(n) = c_1 n + c_2 n\log n \in \theta(n\log n)$. Faltaría calcular las constantes o al menos asegurarnos de que c_2 es positiva, pero calculándolas queda $c_1 = a$ y $c_2 = b$.

Como hemos supuesto que n es potencia de dos habría que aplicar el teorema 2.1 para quitar esa restricción.

2.3.6 Transformación de la imagen

Se utiliza en algunos casos para resolver ecuaciones recurrentes no lineales.

Ejemplo 2.11 Si

$$t(n) = \begin{cases} 6 & si\ n = 1 \\ nt^2\left(\frac{n}{2}\right) & si\ n > 1 \end{cases}$$

haciendo el cambio $n = 2^k$ obtenemos:

$$t\left(2^k\right) = \begin{cases} 6 & si\ k = 0 \\ 2^k t^2\left(2^{k-1}\right) & si\ k > 0 \end{cases}$$

y tomando logaritmos:

$$\log t_k = \begin{cases} \log 6 & si\ k = 0 \\ k + 2\log t_{k-1} & si\ k > 0 \end{cases}$$

Se hace una transformación de la imagen $v_k = \log t_k$, con lo que queda:

$$v_k = \begin{cases} \log 6 & si\ k = 0 \\ k + 2v_{k-1} & si\ k > 0 \end{cases}$$

La ecuación característica es $(x-2)(x-1)^2 = 0$, por lo que las soluciones posibles son de la forma $v_k = c_1 2^k + c_2 + c_3 k$. Se calculan c_1, c_2 y c_3 planteando las ecuaciones correspondientes a v_0, v_1 y v_2, con lo que queda $v_k = (3+\log 3)2^k - k - 2 = \log t_k$, y tomando exponentes y simplificando queda que $t(n) = \frac{2^{3n-2}3^n}{n}$.

Como hemos supuesto que n es potencia de dos habría que aplicar el teorema 2.1 para quitar esa restricción.

2.4 Problemas

Problema 2.1 Obtener O y Ω para el algoritmo de multiplicación de matrices:

```
FOR i = 1 TO n
   FOR j = 1 TO n
      suma = 0
      FOR k = 1 TO n
         suma = suma + a[i,k]b[k,j]
      ENDFOR
      c[i,j] = suma
```