7. Funciones recursivas

Para analizar el tiempo de ejecución de un procedimiento recursivo, le asociamos una función de eficiencia desconocida, T(n), y la estimamos a partir de T(k) para distitnos valores de k.

Ejemplo:

```
1:  int fact(int n)
2:  {
3:    int n;
4:    if (n ≤ 1)
5:        return 1;
6:      else
7:        return (n * fact(n -1));
8:  };
```

Llamamos T(n) al tiempo de ejecución de `fact(n)`. Las líneas 4 y 5 son operaciones elementales, sean c y d, respectivamente sus tiempos de ejecución. Entonces

$$\mathrm{T}(n) = \begin{cases} c + \mathrm{T}(n-1), & \text{si } n > 1 \\ d, & \text{si } n \leq 1 \end{cases}$$

De forma iterativa tendremos:

$$\begin{aligned} \mathrm{T}(n) &= c + \mathrm{T}(n-1) = \\ &= c + (c + \mathrm{T}(n-2)) = 2c + \mathrm{T}(n-2) = \\ &= 2c + (c + \mathrm{T}(n-3)) = 3c + \mathrm{T}(n-3) = \\ &\quad \dots \\ &= ic + \mathrm{T}(n-i) = \\ &\quad \dots \\ &= (n-1)\,c + \mathrm{T}(n-(n-1)) = (n-1)c + d \end{aligned}$$

De donde T(n) es O(n).

$$T(n) = \begin{cases} 1 + T(n-1) & n > 1 \\ 1 & n \leq 1 \end{cases}$$

$$\boxed{T(n) = 1 + T(n-1) \quad n > 1 \qquad T(1) = T(0) = 1}$$

$$\left. \begin{array}{l} T(n) = 1 + \underbrace{T(n-1)}_{\substack{\shortparallel \\ 1+T(n-2)}} \quad n > 1 \\ \qquad\qquad n > 2 \end{array} \right\} \rightarrow \begin{array}{l} 1 + 1 + T(n-2) \quad n > 2 \\ 2 + T(n-2) \quad n > 2 \end{array}$$

$$\left. \begin{array}{l} T(n) = 2 + \underbrace{T(n-2)}_{\substack{\shortparallel \\ 1+T(n-3)}} \quad n > 2 \\ \qquad\qquad n > 3 \end{array} \right\} \rightarrow \begin{array}{l} 2 + 1 + T(n-3) \quad n > 3 \\ 3 + T(n-3) \quad n > 3 \end{array}$$

$$\underline{T(n) = 3 + T(n-3) \quad n > 3}$$

En general:

$$T(n) = i + T(n-i) \quad n > i$$

y para $i = n-1$

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

$$= n-1+1 = n$$

$$T(n) = n \longrightarrow T(n) \text{ es } \underline{\underline{O(n)}}$$

```c
void ordena (int matriz[], int n)
{
    register i;
    int maxi;
    if (n>1) {
        maxi=0;
        for (i=1; i<n; i++)
            if (matriz[i] > matriz[maxi])
                maxi=i;
        if (maxi != 0) {
            i=matriz[0];
            matriz[0]= matriz[maxi];
            matriz[maxi]=i;
        }
        ordena (matriz+1, n-1);
    }
}
```

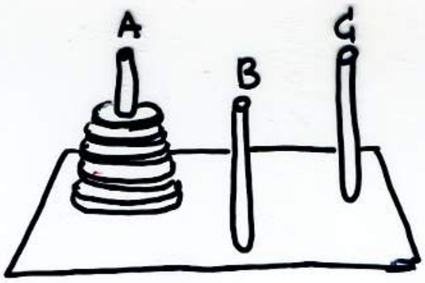
$$T(n) = \begin{cases} n + T(n-1) & n \geq 2 \\ 1 & n = 1 \end{cases}$$

```
void fractal (int n, int cx, int cy, int t)
{
  if (n>0) {
      rectangle (cx-t/2, cy-t/2, cx+t/2, cy+t/2);
      fractal (n-1, cx-t/2, cy-t/2, t/4);
      fractal (n-1, cx+t/2, cy+t/2, t/4);
      fractal (n-1, cx-t/2, cy+t/2, t/4);
      fractal (n-1, cx+t/2, cy-t/2, t/4);
  }
```

/* rectangle (a, b, c, d) dibuja un rectangulo con esquina superior izqda (a, b) y esquina inferior drcha (c, d) */ (cte)

$$T(n) = \begin{cases} 1 + 4T(n-1) & n \geq 1 \\ 1 & n = 0 \end{cases}$$

Otro ejemplo: las torres de Hanoi



Meta: Transferir los N discos del palo A al C

Reglas: . Mover un disco cada vez

. Nunca situar un disco mayor sobre uno de menor tamaño

Solución recursiva

- Transferir N−1 discos de A hasta B
- Mover el disco mayor de A hasta C
- Transferir N−1 discos de B hasta C

Nº total de movimientos

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

Ecuación recurrente a resolver:

$$T(n) = \begin{cases} 2T(n-1) + 1 & n \geq 1 \\ 1 & n = 1 \end{cases}$$

Solución

Expandir la recurrencia

$$T(n) = 2\underbrace{T(n-1)}_{=\,2T(n-2)+1} + 1 \qquad n > 1$$

$$= 2^2 \underbrace{T(n-2)}_{=\,2T(n-3)+1} + 2 + 1 \qquad n > 2$$

$$T(n) = 2^3\, T(n-3) + 2^2 + 2 + 1 \qquad n > 3$$

Y para $n > i$

$$T(n) = 2^i\, T(n-i) + (2^{i-1} + \cdots + 2^2 + 2 + 1)$$

para $\underline{i = n-1}$ alcanzamos el caso base. Así

$$T(n) = 2^{n-1} \underbrace{T(1)}_{=\,1} + (2^{n-2} + \cdots + 2^2 + 2 + 1)$$

$$T(n) = 2^{n-1} + 2^{n-2} + \cdots + 2^2 + 2 + 1 = \frac{2^{n-1} \cdot 2 - 1}{2 - 1} = 2^n - 1$$

Por tanto resulta un tiempo $O(2^n)$

```
int busca (int matriz [], int n, int el)
{ int centro;
  if (n>0) {
        centro = n/2;
        if (matriz [centro] > el)
            return busca (matriz, centro, el);
        else if (matriz [centro] < el)
            return busca (matriz + centro + 1,
                          n - centro - 1, el);
          else return centro;
      }
  else return -1;
}
```

$$T(n) = \begin{cases} 1 + T(n/2) & n>1 \\ 1 & n=1 \end{cases}$$

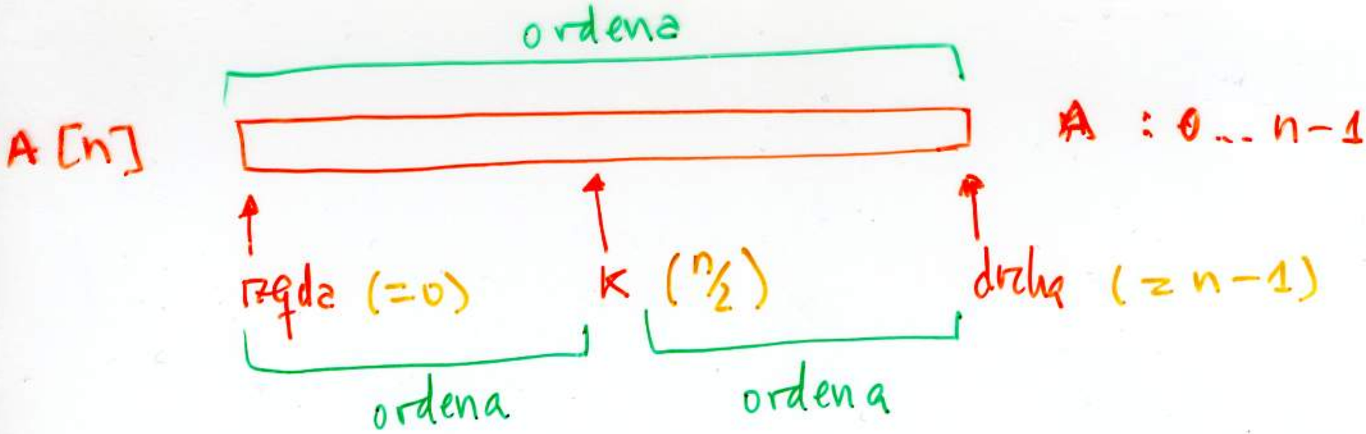
```
int max (int i, n)
  {
   int m1, m2;

   if (n==1)
        return A[i];

   else {
        m1 = max(i, n/2);
        m2 = max (i+n 2, n 2);
        if (m1 < m2)
             return m2;
        else return m1;
        }
  }
```

$$T(n) = \begin{cases} 1 & n = 1 \\ 1 + 2T(n/2) & n > 1 \end{cases}$$

```
void ordena (matriz A, int izqda, drcha)
{
    int k;
    if (izqda > drcha)                        → O(1)
    {
        k= particion (izqda, drcha);          → O(n)
        ordena (A, izqda, k-1);               → T(n/2)
        ordena (A, k+1, drcha);               → T(n/2)
    }
}
```



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

Otra recurrencia:

$$T(n) = 2T(n/2) + n \quad n \geq 2 \quad T(1) = 1$$

2 caminos:

a) expandir la recurrencia:

$$T(n) = 2\underbrace{T(n/2)}_{\shortparallel} + n \quad n \geq 2$$

$$2T(n/4) + n/2 \equiv 2^2 \underbrace{T(n/4)}_{\shortparallel} + 2n \quad n \geq 4$$

$$2T(n/8) + n/4$$

$$T(n) = 2^3 T(n/8) + 3n \quad n \geq 8$$

$$\vdots$$

$$T(n) = 2^i T\left(\frac{n}{2^i}\right) + iN \quad n \geq 2^i$$

y para $i = \log_2 n$ tenemos: $T(n) = n + n \log_2 n$

y por tanto $T(n)$ es $O(n \log_2 n)$

b) Hacer un cambio de variable

$n = 2^m$ $\quad T(2^m) = 2T(2^{m-1}) + 2^m \quad m \geq 1$

$n \geq 2 \equiv 2^m \geq 2^1 \equiv m \geq 1$

y ahora se expande la recurrencia.

Esto se ve mejor en ejemplos más complejos:

$$T(n) = 2T(\sqrt{n}) + 1 \quad T(2) = 0$$

$n = 2^{2^m}$ $\quad T(2^{2^m}) = 2T(2^{2^{m-1}}) + 1 \quad T(2) = 0$

$m \geq 1$

$$\boxed{T(2^m) = 2T(2^{m-1}) + 2^m} \quad m \geq 1 \qquad T(1) = 1$$

$$\underbrace{2T(2^{m-1})}_{=\,\left(2T(2^{m-2}) + 2^{m-1}\right)} \quad m \geq 2$$

$$\left(2T(2^{m-2}) + 2^{m-1}\right) + 2^m \quad m \geq 2$$

$$\equiv \; 2^2 T(2^{m-2}) + 2^m + 2^m = 2^2 T(2^{m-2}) + 2 \cdot 2^m$$

$$\boxed{T(2^m) = 2^2 T(2^{m-2}) + 2 \cdot 2^m \quad m \geq 2}$$

$$\left(2T(2^{m-3}) + 2^{m-2}\right) \quad m \geq 3$$

$$\equiv \; 2^3 T(2^{m-3}) + 2^m + 2 \cdot 2^m = 2^3 T(2^{m-3}) + 3 \cdot 2^m$$

$$\boxed{T(2^m) = 2^3 T(2^{m-3}) + 3 \cdot 2^m \quad m \geq 3}$$

En general para $m \geq i$:

$$\boxed{T(2^m) = 2^i T(2^{m-i}) + i\, 2^m \quad m \geq i}$$

y para $m = i$

$$T(2^m) = 2^m T(1) + m\, 2^m = 2^m + m\, 2^m$$

y como $n = 2^m \Rightarrow m = \log_2 n$

$$T(n) = n + n \log_2 n$$ con lo que:

$$T(n) \text{ es } O(n \log_2 n)$$

```
void reglar (int izda, drcha, h)
{
  int mitad;
  if (h > 0)
    {
      mitad = (izda + drcha) / 2;            → O(1)
*       marcar (mitad, h);                   → O(1)
        reglar (izda, mitad, h-1);           → T(h-1)
        reglar (mitad, drcha, h-1);          → T(h-1)
    }
}
```

$$T(h) = \begin{cases} 1 & h = 0 \\ 1 + 2T(h-1) & h \geq 1 \end{cases}$$

$[\, T(1) = 1 + 2T(0) = 1 + 2 = 3 \qquad T(1) = 3 \,]$

$$\boxed{T(h) = 1 + 2T(h-1) \quad h \geq 1 \qquad T(0) = 1}$$

$T(h)$ es $O(2^h)$

Veamos p. ej. cómo funciona reglar (0, 8, 3)

- reglar (0, 8, 3)
  - h = 3; mitad = 4;
  - marcar (4, 3)
  - reglar (0, 4, 2)
    - h = 2; mitad = 2;
    - marcar (2, 2);
    - reglar (0, 2, 1)
      - h = 1; mitad = 1;
      - marcar (1, 1)
      - reglar (0, 1, 0) → h=0 FIN
      - reglar (1, 2, 0) → h=0 FIN
    - reglar (2, 4, 1)
      - h = 1; mitad = 3;
      - marcar (3, 1)
      - reglar (2, 3, 0) → h=0 FIN
      - reglar (3, 4, 0) → h=0 FIN
  - reglar (4, 8, 2)
    - h = 2; mitad = 6;
    - marcar (6, 2)
    - reglar (4, 6, 1)
      - h = 1 ; mitad = 5;
      - marcar (5, 1)
      - reglar (4, 5, 0) → h=0 FIN
      - reglar (5, 6, 0) → h=0 FIN
    - reglar (6, 8, 1)
      - h = 1 ; mitad = 7;
      - marcar (7, 1)
      - reglar (6, 7, 0) → h=0 FIN
      - reglar (7, 8), 0) → h=0 FIN

$$T(n) = T(n/2) + n^2 \quad n \geq 2 \quad T(1) = 1$$

$$n = 2^m \Rightarrow n^2 = (2^m)^2 = 2^{2m} = 4^m \qquad n \geq 2 \Leftrightarrow 2^m \geq 2^1 \Leftrightarrow m \geq 1$$

$$* \; T(2^m) = \underbrace{T(2^{m-1})}_{=\, T(2^{m-2}) + 4^{m-1} \;\; m \geq 2} + 4^m \quad m \geq 1 \;\Bigg\}\; = T(2^{m-2}) + 4^{m-1} + 4^m \quad m \geq 2$$

$$* \; T(2^m) = \underbrace{T(2^{m-2})}_{=\, T(2^{m-3}) + 4^{m-2} \;\; m \geq 3} + 4^{m-1} + 4^m \quad m \geq 2 \;\Bigg\}\; = T(2^{m-3}) + 4^{m-2} + 4^{m-1} + 4^m \quad m \geq 3$$

$$* \; T(2^m) = T(2^{m-3}) + 4^{m-2} + 4^{m-1} + 4^m \quad m \geq 3$$

En general: $(m \geq i)$

$$T(2^m) = T(2^{m-i}) + [4^{m-(i-1)} + \dots + 4^{m-2} + 4^{m-1} + 4^m]$$

$\boxed{i = m}$

$$T(2^m) = T(1) + [4^1 + \dots + 4^{m-2} + 4^{m-1} + 4^m] =$$

$$= 4^0 + 4^1 + \dots + 4^{m-2} + 4^{m-1} + 4^m =$$

$$= \frac{4 \cdot 4^m - 1}{4 - 1} = \frac{4}{3} 4^m - \frac{1}{3} \qquad [4^m = n^2]$$

$$= \frac{4}{3} n^2 - \frac{1}{3}$$

$T(n)$ es $O(n^2)$

$$T(n) = T(\sqrt{n}) + \log_2 \log_2 n + \log_2 n \quad n \geqslant 4 \quad T(2) = 1$$

$$n = 2^{2^m} \quad * \; T(2^{2^m}) = \underbrace{T(2^{2^{m-1}})}_{} + m + 2^m \quad m \geqslant 1$$

$$\overset{\shortparallel}{T(2^{2^{m-2}})} + (m-1) + 2^{m-1} \quad m \geqslant 2$$

$$* \; T(2^{2^m}) = \underbrace{T(2^{2^{m-2}})}_{} + [m + (m-1)] + [2^m + 2^{m-1}] \quad m \geqslant 2$$

$$\overset{\shortparallel}{T(2^{2^{m-3}})} + (m-2) + 2^{m-2} \quad m \geqslant 3$$

$$* \; T(2^{2^m}) = T(2^{2^{m-3}}) + [m + (m-1) + (m-2)] + [2^m + 2^{m-1} + 2^{m-2}] \quad m \geqslant 3$$

En general:

$$T(2^{2^m}) = T(2^{2^{m-i}}) + [m + (m-1) + \dots + (m-(i-1))] + [2^m + 2^{m-1} + \dots + 2^{m-(i-1)}] \quad m \geqslant i$$

$\boxed{i = m}$

$$T(2^{2^m}) = T(2^{2^0}) + [m + (m-1) + \dots + 1] + [2^m + 2^{m-1} + \dots + 2^1] =$$

$$= 1 + \left[\frac{m+1}{2} \cdot m\right] + \left[\frac{2 \cdot 2^m - 2}{2 - 1}\right] =$$

$$= 1 + \tfrac{1}{2} m^2 + \tfrac{1}{2} m + 2 \cdot 2^m - 2 \quad \left[\begin{array}{l} n = 2^{2^m} \\ \log_2 n = 2^m \\ \log_2 \log_2 n = m \end{array}\right] =$$

$$= \tfrac{1}{2} (\log_2 \log_2 n)^2 + \tfrac{1}{2} \log_2 \log_2 n + 2 \cdot \log_2 n - 1$$

$T(n)$ es $\underline{O(\log_2 n)}$

$$T(n) = 2T(\sqrt{n}) + \log_2 n \qquad n \geq 4 \qquad T(2)=1$$

$$\boxed{n = 2^{2^m}}$$

$$T(2^{2^m}) = 2T(2^{2^{m-1}}) + 2^m \qquad m \geq 1$$

$$\left[n = 2^{2^m} \geq 4 = 2^{2^1} \iff 2^m \geq 2^1 \iff m \geq 1 \right]$$

$$* \quad T(2^{2^m}) = 2\underbrace{T(2^{2^{m-1}})}_{\substack{\| \\ 2T(2^{2^{m-2}}) + 2^{m-1} \quad m \geq 2}} + 2^m \quad m \geq 1 \quad = 2^2 T(2^{2^{m-2}}) + 2 \cdot 2^m \quad m \geq 2$$

$$* \quad T(2^{2^m}) = 2^2 \underbrace{T(2^{2^{m-2}})}_{\substack{\| \\ 2T(2^{2^{m-3}}) + 2^{m-2} \quad m \geq 3}} + 2 \cdot 2^m \quad m \geq 2 \quad = 2^3 T(2^{2^{m-3}}) + 3 \cdot 2^m \quad m \geq 3$$

$$* \; T(2^{2^m}) = 2^3 T(2^{2^{m-3}}) + 3 \cdot 2^m \qquad m \geq 3$$

En general para $m \geq i$ tenemos

$$* \; T(2^{2^m}) = 2^i T(2^{2^{m-i}}) + i \cdot 2^m \qquad m \geq i$$

y para $m = i$:

$$T(2^{2^m}) = 2^m T(2^{2^{m-m}}) + m \cdot 2^m =$$

$$= 2^m T(2) + m \cdot 2^m = 2^m + m\, 2^m$$

Deshaciendo el cambio: $\quad n = 2^{2^m} \Rightarrow m = \log_2 \log_2 n$

$$2^m = \log_2 n$$

Por tanto:

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

Así $T(n)$ es $\underline{\underline{O(\log_2 n \log_2 \log_2 n)}}$

$$T(n) = 2T(n/2) + n^3 \qquad n \geqslant 2 \qquad T(1) = 1$$

$$n = 2^m \Rightarrow n^3 = 8^m \quad [(2^m)^3 = 2^{3m} = 2^{3^m} = 8^m] \qquad 2^m \geqslant 2^1 \Leftrightarrow m \geqslant 1$$

$$T(2^m) = 2\underbrace{T(2^{m-1})}_{= 2T(2^{m-2}) + 8^{m-1}} + [8^m] \quad m \geqslant 1 \quad \Bigg\} = 2^2 T(2^{m-2}) + 2 \cdot 8^{m-1} + 8^m \quad m \geqslant 2$$

$$T(2^m) = 2^2 \underbrace{T(2^{m-2})}_{= 2T(2^{m-3}) + 8^{m-2}} + [2 \cdot 8^{m-1} + 8^m] \quad m \geqslant 2 \quad \Bigg\} = 2^3 T(2^{m-3}) + 2^2 8^{m-2} + 2^1 8^{m-1} + 2^0 8^{m-0} \quad m \geqslant 3$$

$$T(2^m) = 2^3 T(2^{m-3}) + [2^2 8^{m-2} + 2^1 8^{m-1} + 2^0 8^{m-0}] \quad m \geqslant 3$$

En general, para $m \geqslant i$ tenemos:

$$T(2^m) = 2^i T(2^{m-i}) + [2^{i-1} 8^{m-(i-1)} + \cdots + 2^2 8^{m-2} + 2^1 8^{m-1} + 2^0 8^{m-0}]$$

y para $i = m$:

$$T(2^m) = 2^m \underbrace{T(1)}_{=1} + [2^{m-1} 8^1 + \cdots + 2^2 8^{m-2} + 2^1 8^{m-1} + 2^0 8^m]$$

Por tanto:

$$T(2^m) = 2^m 8^0 + 2^{m-1} 8^1 + 2^{m-2} 8^2 + \cdots + 2^2 8^{m-2} + 2^1 8^{m-1} + 2^0 8^m =$$

$$= 2^m + 2^{m-1} 2^3 + 2^{m-2} (2^3)^2 + \cdots + 2^2 (2^3)^{m-2} + 2 (2^3)^{m-1} + (2^3)^m =$$

$$= 2^m + 2^{m+2} + 2^{m+4} + \cdots + 2^{3m-4} + 2^{3m-2} + 2^{3m} =$$

[Progresión geométrica de razón $2^2 = 4$]

$$= \frac{2^{3m} \cdot 4 - 2^m}{4 - 1} = \frac{4}{3} (2^m)^3 - \frac{1}{3} 2^m \quad [2^m = n] =$$

$$= \frac{4}{3} n^3 - \frac{1}{3} n \Rightarrow T(n) \text{ es } \underline{O(n^3)}$$