Análisis de Algoritmos (cont.)

Agustín J. González

ELO320 1º sem 2002

Estrategia: Dividir y Conquistar

- Muchos algoritmos naturalmente tienen una estructura recursiva. Su estructura general es:
  - Dividir el problema en un número de subproblemas
  - Conquistar el problema resolviendolo recursivamente. Si el problema es pequeño se resuelve en forma directa.
  - Combinar las soluciones de los subproblemas.
- Ejemplo: merge-sort : ordenar n elementos
  - Pasos:
    - Dividir la secuencia de n elementos en dos subsecuencias de tamaño n/2
    - Ordenar las subsecuencias recursivamente usando merge-sort
    - Combinar las soluciones parciales.

Algoritmo Merge-sort

- Sea A un arreglo de n elementos y p, r índices del rango a ordenar.
- Merge-Sort(A, p, r)

```
if ( p < r ) {
      q = parteEntera((p+r)/2);
      Merge-Sort(A, p, q);
      Merge-Sort(A, q+1,r);
      Merge(A, p, q, r);
}
```

Algoritmo Merge-sort

- Sea A un arreglo de n elementos y p, r índices del rango a ordenar.
- Merge-Sort(A, p, r)

```
Merge-Sort(A, p, r)                          --> T(n)
if ( p < r ) {                               --> Θ(1)
      q = parteEntera((p+r)/2);              --> Θ(1)
      Merge-Sort(A, p, q);                   --> T(n/2)
      Merge-Sort(A, q+1,r);                  --> T(n/2)
      Merge(A, p, q, r);                     --> Θ(n)
}
```

¿Cuál es el costo de este algoritmo?

Costo Merge-Sort

$$T(n) = \begin{cases} \Theta(1) & \text{if } n = 1 \\ T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + \Theta(n) & \text{if } n > 1 \end{cases}$$

- ¿Cómo se resuelve esta recurrencia?

Teorema Maestro (Master Theorem)

- Sea a≥1 y b ≥1 constantes, sea f(n) una función y sea T(n)=aT(n/b) +f(n) donde n/b es $\lfloor n/b \rfloor$ o $\lceil n/b \rceil$. Entonces T(n) puede ser acotada asintólicamente por

1. Si $\mathrm{f(n)} = \mathrm{O}(\mathrm{n}^{\log_b a - \varepsilon})$ para algún $\varepsilon > 0$,
   entonces $\mathrm{T(n)} = \Theta(\mathrm{n}^{\log_b a})$
2. Si $\mathrm{f(n)} = \Theta(\mathrm{n}^{\log_b a})$, entonces $\mathrm{T(n)} = \Theta(\mathrm{n}^{\log_b a} \lg n)$
3. Si $\mathrm{f(n)} = \Omega(\mathrm{n}^{\log_b a + \varepsilon})$ para algún $\varepsilon > 0$, y
   si $a\, f(n/b) \leq c\, f(n)$ para
   alguna constante $\mathrm{c} < 1$ y n suficientemente grandes,
   entonces $\mathrm{T(n)} = \Theta(\mathrm{f(n)})$.

1 Uso del teorema maestro

1.Si $\mathrm{f(n)} = \mathrm{O}(\mathrm{n}^{\log_b a - \varepsilon})$ para algún $\varepsilon > 0$ $\Rightarrow \mathrm{T(n)} = \Theta(\mathrm{n}^{\log_b a})$

2.Si $\mathrm{f(n)} = \Theta(\mathrm{n}^{\log_b a})$ $\Rightarrow$ $\mathrm{T(n)} = \Theta(\mathrm{n}^{\log_b a} \lg n)$

3.Si $\mathrm{f(n)} = \Omega(\mathrm{n}^{\log_b a + \varepsilon})$ para algún $\varepsilon > 0$, y

si $a\, f(n/b) \leq c\, f(n)$ para

alguna constante $c < 1$ y suficientemente grandes n,

$\Rightarrow$ $\mathrm{T(n)} = \Theta(\mathrm{f(n)})$.

$$T(n) = 9T(n/3) + n$$

$$a = 9, \quad b = 3, \quad f(n) = n \quad \Rightarrow \quad n^{\log_b a} = n^{\log_3 9} = \Theta(n^2)$$

$$\therefore\ f(n) = \mathrm{O}(n^{\log_3 9 - \varepsilon}), \varepsilon = 1$$

$$\therefore Caso 1 : T(n) = \Theta(n^2)$$

2 Uso del teorema maestro

1.Si $f(n) = O(n^{\log_b a - \varepsilon})$ para algún $\varepsilon > 0 \Rightarrow T(n) = \Theta(n^{\log_b a})$

2.Si $f(n) = \Theta(n^{\log_b a})$ $\Rightarrow T(n) = \Theta(n^{\log_b a} \lg n)$

3.Si $f(n) = \Omega(n^{\log_b a + \varepsilon})$ para algún $\varepsilon > 0$, y

si $a\, f(n/b) \le c\, f(n)$ para

alguna constante $c < 1$ y suficientemente grandes n,

$\Rightarrow \quad T(n) = \Theta(f(n))$.

$$T(n) = T(2n/3) + 1$$

$$a = 1, \quad b = 3/2, \quad f(n) = 1 \quad \Rightarrow \quad n^{\log_b a} = n^{\log_{3/2} 1} = n^o = 1$$

$$\therefore Caso 2 : T(n) = \Theta(\lg n)$$

3 Uso del teorema maestro

$1.\text{Si } \ \mathrm{f(n)} = \mathrm{O}(\mathrm{n}^{\log_b a - \varepsilon}) \ \text{ para algún } \varepsilon > 0 \ \Rightarrow \mathrm{T(n)} = \Theta(\mathrm{n}^{\log_b a})$

$2.\text{Si } \mathrm{f(n)} = \Theta(\mathrm{n}^{\log_b a}) \qquad \Rightarrow \ \mathrm{T(n)} = \Theta(\mathrm{n}^{\log_b a} \lg n)$

$3.\text{Si } \mathrm{f(n)} = \Omega(\mathrm{n}^{\log_b a + \varepsilon}) \text{ para algún } \varepsilon > 0, \text{ y}$

si $a\, f(n/b) \leq c\, f(n)$ para

alguna constante $c < 1$ y suficientemente grandes n,

$\Rightarrow \ \ \mathrm{T(n)} = \Theta(\mathrm{f(n)}).$

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

$$a = 2, \quad b = 2, \quad f(n) = n \quad \Rightarrow \quad n^{\log_b a} = n^{\log_2 2} = n^1 = \mathrm{n}$$

$$\mathrm{f(n)} = \Theta(\mathrm{n}^{\log_b a}) \quad \Rightarrow \quad Caso 2$$

$$\mathrm{T(n)} = \Theta(\mathrm{n}\lg(\mathrm{n}))$$