UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria Informàtica etsinf

Tema 2. Análisis

Programación (PRG)

Jorge González Mollá

Departamento de Sistemas Informáticos y Computación

Índice

1. Introducción
2. Complejidad
3. Casos
4. Recursividad
5. Ordenación
6. Otros algoritmos

Coste Recursivo

- En la complejidad temporal de un algoritmo recursivo influyen:
  - El número de llamadas recursivas generadas por el método.
  - La forma en la que disminuye el tamaño del problema n de llamada en llamada. Normalmente, la reducción es del tipo:
    - n – c (siendo c una constante tal que c ≥ 1)
    - n / c (siendo c una constante tal que c > 1)
  - El coste iterativo del resto de operaciones que realiza el método excluyendo la invocación de las llamadas recursivas.

Ecuaciones de Recurrencia

- Para poder analizar la complejidad de un algoritmo recursivo se utilizan las ecuaciones de recurrencia.

- Las ecuaciones de recurrencia permiten indicar el tiempo de ejecución para los distintos casos de un algoritmo recursivo (caso base y caso general).

- Una vez se tienen las ecuaciones de recurrencia, es posible obtener la expresión de la función de coste de varias maneras.

- Una solución para desarrollar y resolver así estas ecuaciones: la técnica de despliegue de recurrencias, a.k.a., de sustitución.

Ejemplo: Factorial

```
/** n>=0 */
int factorial (int n) {
   if (n == 0) { return 1; }
   else { return n * factorial(n - 1); }
}
```

– Talla del problema: el valor del parámetro n.

– Caso base (n = 0). Tiempo de ejecución: T(n) = 1 (1 comparación y 1 retorno se ejecutan como secuencia en 1 unidad de tiempo).

– Caso general (n > 0). Tiempo de ejecución: T(n) = 1 + T(n - 1), siendo el coste iterativo 1, una secuencia de 4 operaciones: comparar `n == 0`, restar `n - 1`, el producto * y el retorno; más el tiempo de hacer 1 llamada a `factorial` de talla `n - 1`.

– Ecuaciones de recurrencia: $$\mathrm{T}(n) = \begin{cases} 1 & n = 0 \\ 1 + \mathrm{T}(n-1) & n > 0 \end{cases}$$

Ejemplo: Factorial

- La técnica de despliegue de recurrencias consiste en:
  – Sustituir las apariciones de T dentro de la ecuación de recurrencia hasta encontrar un patrón general en la expresión resultante basado en el número total de llamadas recursivas, i.

- En el caso del cálculo del `factorial`:

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

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

$= 2 + (1 + T(n-3)) = 3 + T(n-3) = \ldots = i + T(n-i)$

donde i es el número total de llamadas recursivas.

- Es decir, la última llamada $T(n-i)$ es la del caso base, cuya talla es 0:

$$n - i = 0 \Leftrightarrow i = n \rightarrow T(n) = n + T(0) = n + 1$$

- $T(n) = n + 1$

  $T(n) \in \Theta(n)$