Análisis de la complejidad de algoritmos

Alberto Verdejo

Dpto. de Sistemas Informáticos y Computación

Universidad Complutense de Madrid

Octubre 2008

Complejidad en promedio

- Para analizar el coste en el caso promedio necesitamos conocer el tiempo de ejecución de cada posible ejemplar y la frecuencia con que se presenta, es decir, su distribución de probabilidades.
  Definimos la complejidad de un algoritmo A en el caso promedio como

$$TM_A(n) = \sum_{\overline{x} \text{ de tamaño } n} p(\overline{x}) t_A(\overline{x})$$

  siendo $p(\overline{x}) \in [0..1]$ la probabilidad de que la entrada sea $\overline{x}$.
- **NO** se debe interpretar el análisis del caso medio como el análisis del caso típico. Una media solo es típica si los casos no se desvían demasiado de dicha media, es decir si la desviación típica es pequeña.
- Un análisis en el caso medio es útil porque nos dice cuánto tarda el algoritmo cuando se ejecuta **muchas veces sobre entradas muy diferentes**.
- Por ejemplo, el uso de un algoritmo de ordenación que se usa repetidamente para ordenar todas las posibles entradas. Se puede tolerar una ordenación relativamente lenta (para ciertos casos) si en media el tiempo de ordenación es bueno, ej: quicksort
- En sistemas críticos es mejor el análisis del caso peor.

Complejidad en promedio: búsqueda secuencial

fun búsqueda-sec$(V[1..n]$ **de** *ent*, x : *ent*$)$ **dev** i : *nat*
    $i := 1$;
    mientras $i \leq n \ \wedge_c \ V[i] \neq x$ **hacer**
        $i := i+1$
    fmientras
ffun

Caso peor: $\Theta(n)$

Caso mejor: $\Theta(1)$

¿Caso medio?

$$TM_A(n) = \sum_{\overline{x} \text{ de tamaño } n} p(\overline{x}) t_A(\overline{x})$$

Primer análisis: x está con seguridad en V. Todos los elementos son distintos y x puede aparecer en cualquier posición con la misma probabilidad, $\frac{1}{n}$.

Hacemos clases de vectores, según lo que tarde el algoritmo.

Clases

C_1 si $V[1] = x$

C_2 si $V[1] \neq x$ y $V[2] = x$

$\vdots$

C_n si $(\forall i : 1 \leq i < n : V[i] \neq x) \ \wedge \ V[n] = x$

i	probabilidad	comparaciones
1	$1/n$	1
2	$1/n$	2
	$\vdots$	
i	$1/n$	i
	$\vdots$	
n	$1/n$	n

$$TM_A(n) = \frac{1}{n} \sum_{i=1}^{n} i = \frac{n+1}{2}$$

Segundo análisis: x está con probabilidad p, $0 \leq p \leq 1$.

La probabilidad de que esté en la posición i es $\frac{p}{n}$.

Una clase más, $n+1$, $(\forall i : 1 \leq i \leq n : V[i] \neq x)$. Se realizan n comparaciones.

$$\begin{aligned} TM_A(n) &= (\sum_{i=1}^{n} \frac{p}{n} i) + (1-p)n = p\frac{n+1}{2} + (1-p)n \\ &= \frac{(2-p)n+p}{2} \in \Theta(n) \end{aligned}$$

Si $p = 1/2$ entonces $TM_A(n) = 3n/4 + 1/4$, es decir, en media se recorre al menos los $3/4$ del vector.

Complejidad en promedio: ordenación por inserción

```
proc ord-inserción(V[1..n] de ent)
    para i = 2 hasta n hacer
        elem := V[i] ;
        j := i − 1 ;
        mientras j > 0 ∧_c elem < V[j] hacer
            V[j + 1] := V[j] ;
            j := j − 1
        fmientras ;
        V[j + 1] := elem
    fpara
fproc
```

Caso peor: $\Theta(n^2)$

Caso mejor: $\Theta(n)$

¿Caso medio?

Suposiciones:

- Todos los elementos son distintos: $n!$ posibilidades.
- Todas las posibilidades igualmente probables.

1ª posibilidad: Sumar los tiempos de todas las posibilidades y dividir por $n!$.

2ª posibilidad: Analizar el tiempo requerido razonando *probabilísticamente* a medida que vayamos avanzando.

- Definimos el **rango parcial** de $V[i]$ $(2 \leq i \leq n)$ como la posición que ocuparía $V[i]$ si ordenásemos $V[1..i]$.
  Ejemplo: el rango de $V[4]$ en $V = [3,6,2,5,1,7,4]$ es 3.
- El rango parcial de $V[i]$ no depende del orden de los elementos en $V[1..i-1]$.
- Si las $n!$ permutaciones son equiprobables, el rango parcial de $V[i]$ puede variar en $1..i$ con la misma probabilidad para todos los valores.

Fijemos i y supongamos que vamos a entrar en el bucle **mientras** y que el rango parcial de $V[i]$ es k.

Entonces la comprobación del bucle se realiza $i-k+1$ veces.

El número medio de veces que se efectúa la comprobación para cada i es:

$$c_i = \sum_{k=1}^{i} \frac{1}{i}(i-k+1) = \frac{i+1}{2}$$

Por ser sucesos independientes para distintos valores de i, el número total medio de comparaciones es:

$$\sum_{i=2}^{n} c_i = \sum_{i=2}^{n} \frac{i+1}{2} = \frac{(n+4)(n-1)}{4} \in \Theta(n^2)$$

Fórmulas útiles para el análisis de algoritmos

Sumatorios

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2} \approx \frac{1}{2}n^2$$

$$\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6} \approx \frac{1}{3}n^3$$

$$\sum_{i=1}^{n} i^k \approx \frac{1}{k+1}n^{k+1}$$

$$\sum_{i=0}^{n} a^i = \frac{a^{n+1}-1}{a-1} \qquad \text{si } a \neq 1$$

$$\sum_{i=1}^{n} i2^i = (n-1)2^{n+1} + 2$$

$$\sum_{i=1}^{n} \log i \approx n \log n$$

Fórmulas útiles para el análisis de algoritmos

Propiedades de los logaritmos, $a, b > 1$

$$\log_a 1 = 0$$

$$\log_a a = 1$$

$$\log_a x^y = y \log_a x$$

$$\log_a xy = \log_a x + \log_a y$$

$$\log_a \frac{x}{y} = \log_a x - \log_a y$$

$$a^{\log_b x} = x^{\log_b a}$$

$$\log_a x = \frac{\log_b x}{\log_b a}$$

Quicksort

```
proc quicksort(V[1..n] de elem, e c,f : nat)
    si c < f entonces
        partición(V,c,f,p)
        quicksort(V,c,p-1)
        quicksort(V,p+1,f)
    fsi
fproc
```

Quicksort: coste

$$T(n) = \begin{cases} c'_0 & n = 0 \\ T(p) + T(q) + c'_1 n & n > 0 \end{cases}$$

con $p + q = n - 1$.

Cuando se divide siempre por la mitad, la recurrencia se aproxima de la forma

$$T(n) = \begin{cases} c'_0 & n = 0 \\ 2T(n/2) + c'_1 n & n > 0 \end{cases}$$

cuya solución está en $\Theta(n \log n)$ por el teorema de la división.

Cuando el pivote queda sistemáticamente colocado en uno de los extremos, la recurrencia queda

$$T(n) = \begin{cases} c'_0 & n = 0 \\ T(n-1) + c'_0 + c'_1 n & n > 0 \end{cases}$$

cuya solución está en $\Theta(n^2)$ por el teorema de la resta.

¿Puede haber casos peores todavía con otras elecciones de p y q?

Quicksort: caso peor

$T(n) \in O(n^2)$ (para cualquier elección de p y q)

Demostramos mediante *inducción constructiva* que $\forall n : n \geq 1 : T(n) \leq Cn^2$.

En el caso básico $n = 1$ tenemos que

$$T(1) = T(0) + T(0) + c'_1 = 2c'_0 + c'_1 \leq C$$

Supongamos que la propiedad es cierta para cualquier $m < n$.

Entonces, en el paso de inducción tenemos para $n > 1$:

$$T(n) = T(p) + T(q) + c'_1 n \leq^{h.i.} Cp^2 + Cq^2 + c'_1 n.$$

Como $p + q = n - 1$, $p^2 + q^2 = (n-1)^2 - 2pq \leq (n-1)^2$, por lo que

$$T(n) \leq C(n-1)^2 + c'_1 n = Cn^2 - 2Cn + C + c'_1 n.$$

De aquí, $Cn^2 - 2Cn + C + c'_1 n \leq Cn^2 \Leftrightarrow C + (c'_1 - 2C)n \leq 0$.

De acuerdo con la primera restricción, podemos tomar como constante $C = 2c'_0 + c'_1$, y comprobamos que esta elección satisface también la última restricción:

$$(2c'_0 + c'_1) + (c'_1 - 2(2c'_0 + c'_1))n = (2c'_0 - 4c'_0 n) + (c'_1 - c'_1 n) \overset{n \geq 1}{\leq} 0 + 0 = 0$$

2

Basta tomar $C = 2c'_0 + c'_1$ para que $\forall n : n \geq 1 : T(n) \leq Cn^2$.

Quicksort: caso medio

El caso medio está determinado por la variación en los tamaños de los vectores con los que se hacen las llamadas recursivas, p y q. Ambos pueden variar de 0 a $n-1$, pero siempre cumpliendo $p+q=n-1$, o $q=n-p-1$.

$$
\begin{aligned}
T_m(n) &= \frac{1}{n}\sum_{p=0}^{n-1}\left(T_m(p)+T_m(n-p-1)+(n-1)\right) \\
&= \frac{1}{n}\left(\sum_{p=0}^{n-1}T_m(p)+\sum_{p=0}^{n-1}T_m(n-p-1)+\sum_{p=0}^{n-1}(n-1)\right) \\
&= \frac{1}{n}\left(\sum_{p=0}^{n-1}T_m(p)+\sum_{j=0}^{n-1}T_m(j)\right)+(n-1) \\
&= (n-1)+\frac{2}{n}\sum_{p=0}^{n-1}T_m(p) \\
&= (n-1)+\frac{2}{n}\sum_{p=2}^{n-1}T_m(p)
\end{aligned}
$$

donde el último paso se puede dar ya que $T_m(0)=T_m(1)=0$.

Es una recurrencia con historia. La resolvemos calculando el término para n, para $n-1$ y restando. La recurrencia para n es

$$nT_m(n) = n(n-1) + 2\sum_{p=2}^{n-1} T_m(p)$$

para $n-1$,

$$(n-1)T_m(n-1) = (n-1)(n-1-1) + 2\sum_{p=2}^{n-2} T_m(p)$$

y restando,

$$\begin{aligned} nT_m(n) - (n-1)T_m(n-1) &= 2T_m(n-1) + 2(n-1) \\ nT_m(n) &= (n+1)T_m(n-1) + 2(n-1) \end{aligned}$$

Dividiendo por $n(n+1)$, queda:

$$\frac{T_m(n)}{n+1} = \frac{T_m(n-1)}{n} + \frac{2(n-1)}{n(n+1)}$$

Definimos

$$S(n) = \frac{T_m(n)}{n+1} \qquad S(n) = \begin{cases} 0 & n \leq 1 \\ S(n-1) + \frac{2(n-1)}{n(n+1)} & n \geq 2 \end{cases}$$

y ya que se cumple $1 > \frac{n-1}{n+1}$ podemos simplificar $S(n)$,

$$S(n) \leq \begin{cases} 0 & n \leq 1 \\ S(n-1) + \frac{2}{n} & n \geq 2 \end{cases}$$

Ahora, desplegando,

$$\begin{aligned} S(n) &\leq S(n-1) + \frac{2}{n} \\ &\leq S(n-2) + \frac{2}{n-1} + \frac{2}{n} \\ &\leq S(n-3) + \frac{2}{n-2} + \frac{2}{n-1} + \frac{2}{n} \\ &\vdots \\ &\leq S(n-i) + 2 \sum_{j=n-i+1}^{n} \frac{1}{j} \end{aligned}$$

La recurrencia desaparece cuando $n - i = 1$, o $i = n - 1$. En ese caso,

$$S(n) \leq S(1) + 2\sum_{j=2}^{n} \frac{1}{j} = 2\sum_{j=2}^{n} \frac{1}{j} \leq 2\int_{1}^{n} \frac{1}{x}\mathrm{d}x = 2\ln n$$

Y por tanto

$$\begin{aligned} T_m(n) &= (n+1)S(n) \\ &\leq (n+1)2\ln n \\ &= \frac{2}{\log e}(n+1)\log n \\ &\approx 1{,}386(n+1)\log n \\ &\in O(n\log n) \end{aligned}$$