Carlos Sánchez JG1075

1b Problema computacional

Descripción de un conjunto salida que satisfaga un conjunto entrada, tal que los conjuntos puedan ser procesados por una computadora, y que la entrada pueda obtenerse la salida

Ejemplo:

Multiplicación de matrices.

Entrada: $A=(a_{ij})$ y $B=(b_{ij})$ son matrices $n \times n$,

Salida: $\sum_{k=1}^{n} a_{ij} \cdot b_{ij}$ ; $i, j = 1, 2, \ldots, n$

1d Eficiencia de algoritmo

Propiedad del algoritmo, la cual mide la cantidad de recursos computacionales (tiempo y espacio) usado.

Por ejemplo, el insertion sort su tiempo de ejecución es (en el mejor caso):

$$T(n) = an + b = \Theta(n)$$

← Insertion - sort
selection

Invariante

En el inicio de cada for, de las líneas 2-7, el subarreglo A[1 ... j] consistente en los elementos originale en A[1 ... j], pero ordenados.

Inicialización:

Para la primera etapa (iteración) j=1, el arreglo consiste de un solo elemento A[1], lo cual mantiene la invariante cierta.

Mantenimiento

Suponemos se cumple la invariante, por lo que al inicio de la iteración donde $1 \leq j \leq n-1$, se cumple que A[j] se ordena por el cuerpo del bucle esto es:

$$A[j] = \min A[j+1 .. n]$$

Preservándose la invariante

Finalización:

La condición que finaliza el bucle es: $j > A.length - 1 = n-1$, esto es cuando $j = n$, tenenemos que

A [1 ... n] (arreglo ordenado).

∴ Algoritmo es correcto.

2 $$T(n) = C_1 + C_2 n + C_3(n-1) + \sum_{j=2}^{n-1} C_4(n-j+1)$$
$$+ \sum_{j=2}^{n-1} C_5(n-j) + \sum_{j=2}^{n-1} C_6 t_{nj},$$
$$+ C_7(n-1)$$

En el peor caso (arreglo con elementos más grandes al inicio),

$$t_j = n - j$$

Así

$$T(n) = C_1 + C_2 n + C_3 n - C_3 + \frac{1}{2} C_4(n^2 - n - 2) +$$
$$C_5 \frac{1}{2}(n^2 - 3n + 2) + C_6 \frac{1}{2}(n^2 - 3n + n)$$
$$+ C_7 n - C_7$$

Agrupando para constantes a, b, c

$$T(n) = an^2 + bn + c = \Theta(n^2)$$

donde a, b, c dependen de $C_i$

Esto es coste cuadrático.

5b Invariante

Para cada iteración del bucle 2 - 4

$$A[min] \leq A[1..i-1]$$

Inicialización:

Primera iteración $i=2$, entonces

$$A[1..1] = A[1]$$

trivialmente $A[min] \leq A[1]$, $A[1] \leq A[1]$

La invariante es cierta aquí.

Mantenimiento

Suponemos cierta la invariante cierta para cada iteración, en el cuerpo código tenemos que min=i si $A[i] < A[min]$
es decir

$$A[min] \leq A[1..i]$$

Cumpliéndose la invariante: $A[min] \leq A[1..i-1]$ en la sig. iteración

Finalización:

Al finalizar el ciclo $i=n+1$
es decir

$$A[min] \leq A[1...i-1]$$
$$= A[1...n]$$

∴ La invariante es cierta. Algoritmo correcto

$$T(n) = c_1 + \sum_{i=2}^{n+1} c_2 + \sum_{i=2}^{n} c_3 + \left(\sum_{i=2}^{n} c_4 t_i + c_5\right.$$

$$T(n) = c_1 + c_2 n + c_3(n-1) + c_4 \sum_{i=2}^{n} t_i + c_5$$

En el mejor caso significa que

$A[1] < A[2 \ldots n]$

Entonces

$t_i = 0$ (jamás entra al ciclo)

Obtenemos

$$T(n) = c_1 + c_2 n + c_3 n - c_3 + c_5$$

Agrupando por variables a, b que dependen de $c_i$

$$T(n) = an + b$$

$$¿\, an + b = \Theta(n)?$$

$\exists\, c_1, c_2$ y $\forall n \geq n_0$:

$$0 \leq c_1 n \leq an + b \leq c_2 n$$

Como $n_0 = 1$

$a + \frac{b}{n} \in (a, a+b)$, $c_1 \leq a$ y $c_2 \geq a+b$

Se cumple que

$$an + b = \Theta(n)$$

4b

$$¿\ 27n^2 - 9n + 3 = \Theta(n^2)\ ?$$

Si $\exists\ c_1, c_2 \in \mathbb{R}^+$ y $\forall n \geq n_0$:

$$0 \leq c_1 n^2 \leq 27n^2 - 9n + 3 \leq c_2 n^2$$

$$c_1 \leq 27 - \frac{9}{n} + \frac{3}{n^2} \leq c_2$$

Como $n_0 = 1$

$$27 - \frac{9}{n} + \frac{3}{n^2} \in (21, 27]$$

Tenemos que

$$0 < c_1 \leq 21$$

$$c_2 \geq 27$$

Se cumple que

$$27n^2 - 9n + 3 = \Theta(n^2)$$

4d $¿2n^2+8n-16 = \Omega(n^3)?$

Si: $\exists\, c_1 \in \mathbb{R}^+$ y $\forall n \geq n_0$

$$0 \leq c_1 n^3 \leq 2n^2+8n-16$$

$$c_1 n \leq 2 + \frac{8}{n} - \frac{16}{n^2}$$

Por $n_0 = 1$

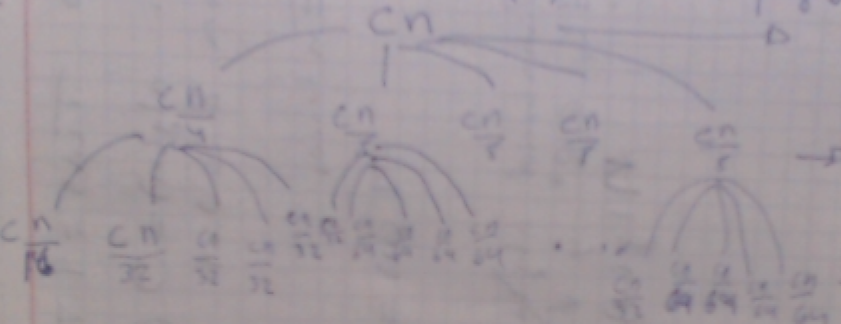
$$2 + \frac{8}{n} - \frac{16}{n^2} \in [2, \infty)$$

Lo cual es una contradicción, $c_1$ debe estar acotado.

Así

$$2n^2+8n-16 \neq \Omega(n^3)$$

5.a $T(n) = T\left(\frac{n}{4}\right) + 4T\left(\frac{n}{8}\right) + n$



$\to cn$

$\to \frac{cn}{4} + \frac{4}{8}cn = \frac{3}{4}cn$

$\to \frac{cn}{16} + \frac{8}{32}cn + \frac{16}{64}cn = \frac{9}{16}cn$

$\to \left(\frac{3}{4}\right)^i cn$

$\log_4(n)$

La ruta más larga de la raíz a las hojas es $n \to \frac{1}{4}n \to \left(\frac{1}{4}\right)^2 n \to \cdots \to 1$, desde $\left(\frac{1}{4}\right)^k n = 1$ cuando $k = \log_4(n)$, la altura del árbol

$T(n) = cn\sum_{i=0}^{\log_4 n}\left(\frac{3}{4}\right)^i = 4nc - c3^{\log_4(n)} < 4nc = O(n)$

$T(n) = O(n)$

5b $T(n) = 9T\left(\frac{n}{3}\right) + n\lg(n)$

Si $n\lg(n) = O\left(n^{\log_3(9) - \varepsilon}\right)$

Para $\varepsilon \simeq 0.4$

Entonces

$T(n) = \Theta\left(n^{\log_3(9)}\right)$

6. Análisis amortizado

Sea $C_i$ el costo de operación $i$:

$$C_i = \begin{cases} i & \text{si } i \text{ es potencia exacta de 3} \\ 1 & \text{de otra manera} \end{cases}$$

Entonces

$$\sum_{i=1}^{n} C_i = \sum_{i=1}^{\lceil \log n \rceil} 3^i + \sum_{i \le n \text{ si } i \text{ no es potencia de } 3} 1$$

$$\le \sum_{i=1}^{\lceil \log n \rceil} 3^i + n$$

$$\le \frac{3}{2} 3^{\lfloor \log_3 (n) \rfloor} - \frac{3}{2} + n$$

$$< \frac{3}{2} n - \frac{3}{2} + n$$

$$= \frac{5}{2} n - \frac{3}{2}$$

$$= O(n)$$

$\Rightarrow \frac{O(n)}{n} = O(1)$, el costo amortizado.

6. Método de contabilización.

Sea

$$c_i = \begin{cases} i & \text{si } i \text{ potencia exacta de } 3 \\ 1 & \text{de otra manera} \end{cases}$$

Entonces

| $i$ | $c_i'$ | $c_i$ | $b_i$ |
|---|---|---|---|
| 1 | 3 | 1 | 2 |
| 2 | 3 | 1 | 4 |
| 3 | 3 | 3 | 4 |
| 4 | 3 | 1 | 6 |
| 5 | 3 | 1 | 8 |
| 6 | 3 | 1 | 10 |
| 7 | 3 | 1 | 12 |
| 8 | 3 | 1 | 14 |
| 9 | 3 | 9 | 8 |

$\vdots$

6. Método de contabilización

Si $i$ es potencia exacta de 3 el costo amortizado $\hat{c}_i$ sin $b_i$ (suma de $c_i$ - $\Delta\Phi_i$)

$$bi - \sum_{j=1}^{\log_b(i)} (b^j - b^{j-1} - 1) - i - \sum_{j=1}^{\log_b(i)-1} b^j =$$

$$bi - (-\log_b(i) + i - 1) - i - \frac{i-b}{b-1} =$$

$$bi - \frac{i-b}{b-1} + \log_b(i) - 2i + 1$$

Por ejemplo $i = 9$ y $b = 3$

$$27 - 2 + 2 - 18 + 1 > 0$$

¿Para qué queremos esa $f(i) \geq b = 3$?

Si

$$\lim_{i \to \infty} f(i) = \infty$$

Entonces

$c_i' = 3$, funciona como costo amortizado

En caso de $b = 2$

$$\lim_{i \to \infty} f(i) = -\infty$$

Naturalmente $b_i$ no alcanzará en algún punto.

Un $c_i$ más ajustado sería $c_i' = 2{,}7$

Como se demostró anteriormente, eso sugiere

$$\sum_{i=1}^{n} c_i' = 3n$$

Así,

$$\sum_{i=1}^{n} c_i' - \sum_{i=1}^{n} c_i = b_i > 0$$

Obtenemos por tanto que el costo amortizado es $O(1)$, entonces el costo de $n$ operaciones es $O(n)$.

7b

El tiempo de ejecución ~~de~~ insertion sort:

$$T(n) = c_1 n + c_2(n-1) + c_4(n-1) + c_5 \sum_{j=2}^{n} t_j + c_6 \sum_{j=2}^{n} (t_j - 1) + c_7 \sum_{j=2}^{n} (t_j - 1) + c_8(n-1)$$

Sin embargo, necesitamos el caso promedio. Como ya se demostró en la tarea I.6 el número de inversiones esperado

$$E[X] = \frac{n(n-1)}{4}$$

que es el número de intercambios.

Así, el caso promedio

$$T(n) = c_1 n + c_2(n-1) + c_4(n-1) + c_5 \frac{n(n-1)}{4} + c_6\left[\frac{n(n-1)}{4} - 1\right] + c_7\left[\frac{n(n-1)}{4} - 1\right] + c_8(n-1)$$

Si agrupamos tal que a, b, c dependen de $c_i$ correspondiente:

$$T(n) = an^2 + bn + c = \Theta(n^2)$$

O si se quiere

$$T(n) = \underbrace{\Theta(n)}_{\text{externo}} + \underbrace{\Theta(n^2)}_{\text{interno}} = \Theta(n^2)$$