INFORMATICA GENERALE - I-Z

Claudia Malvenuto - Daniele A. Gewurz

Scheda esercizi n. 3: alcune soluzioni

1. Se con $\lfloor x \rfloor$ indichiamo la parte intera inferiore di x, cioè il massimo intero m che sia minore o uguale di x, dimostrare che, presi comunque tre numeri naturali a, b e n maggiori di 0, si ha sempre:
$$\lfloor \lfloor n/a \rfloor / b \rfloor = \lfloor n/ab \rfloor.$$
(Questa uguaglianza è utile perché garantisce che se, nel calcolo del tempo di esecuzione di un algoritmo, si approssima $\lfloor n/a \rfloor$ con n/a in iterazioni successive, l'errore che si accumula rimane limitato.)

   Sol. Il numero $\lfloor n/a \rfloor$ è il quoziente della divisione euclidea tra n e a, cioè è l'unico k tale che $n = k \cdot a + r$ con $0 \leq r \leq a - 1$.

   Quindi $\lfloor \lfloor n/a \rfloor / b \rfloor$ è il numero h tale che $k = \lfloor n/a \rfloor = h \cdot b + r'$, con $0 \leq r' \leq b - 1$, e quindi inserendo nell'uguaglianza precedente si ha: $n = (h \cdot b + r')a + r = h \cdot ab + r' \cdot a + r$. Come conseguenza delle disuguaglianze su r e r', si ha che $0 \leq r' \cdot a + r \leq (b-1) \cdot a + (a-1) = ab - 1$, e quindi h è effettivamente anche il quoziente della divisione euclidea tra n e ab.

5. Supponiamo di avere due algoritmi; il tempo di esecuzione di uno, in funzione di un parametro n è definito ricorsivamente da $T(n) = 5T(n/6) + 2n^2$; quello dell'altro è dato da $T(n) = 7T(n/6) + n$. In entrambi casi il tempo di esecuzione del caso base è $O(1)$. Per n sufficientemente grande, quale dei due algoritmi è più veloce?

   Sol. Studiamo per iterazione le ricorrenze date, esaminandone un paio di iterazioni:

$$\begin{aligned}
T(n) &= 5T\left(\frac{n}{6}\right) + 2n^2 = 5\left(5T\left(\frac{n}{6^2}\right) + 2\left(\frac{n}{6}\right)^2\right) + 2n^2 = 5^2T\left(\frac{n}{6^2}\right) + 2\left(1 + \frac{5}{6^2}\right)n^2 \\
&= 5^2\left(5T\left(\frac{n}{6^3}\right) + 2\left(\frac{n}{6^2}\right)^2\right) + 2\left(1 + \frac{5}{6^2}\right)n^2 = 5^3T\left(\frac{n}{6^3}\right) + 2\left(1 + \frac{5}{6^2} + \left(\frac{5}{6^2}\right)^2\right)n^2 \\
&= \ldots = 5^kT\left(\frac{n}{6^k}\right) + 2\sum_{i=0}^{k-1}\left(\frac{5}{6^2}\right)^i n^2
\end{aligned}$$

   Ponendo $k = \log_6 n$, il primo addendo diventa $5^{\log_6 n}O(1) = n^{\log_6 5}O(1) = O(n^{\log_6 5})$. E maggiorando il secondo addendo con la serie geometrica, tutta l'espressione data è minore di:

$$O(n^{\log_6 5}) + 2n^2\sum_{i=0}^{\infty}\left(\frac{5}{6^2}\right)^i = O(n^{\log_6 5}) + 2n^2\frac{1}{1 - \frac{5}{6^2}} = O(n^{\log_6 5}) + kn^2 = O(n^2),$$

   dato che $\log_6 5 < 2$.

   Per quanto riguarda la seconda ricorrenza:

$$T(n) = 7\,T\left(\frac{n}{6}\right) + n = 7\left(7T\left(\frac{n}{6^2}\right) + \frac{n}{6}\right) + n = 7^2\,T\left(\frac{n}{6^2}\right) + \left(1 + \frac{7}{6}\right)n$$

$$
\begin{aligned}
&= \quad \ldots = 7^k\, T\left(\frac{n}{6^k}\right) + \sum_{i=0}^{k-1}\left(\frac{7}{6}\right)^i n = O(n^{\log_6 7}) + n\frac{1-\left(\frac{7}{6}\right)^{\log_6 n}}{1-\frac{7}{6}} \\
&= \quad O(n^{\log_6 7}) + 6n\left(\left(\frac{7}{6}\right)^{\log_6 n} - 1\right) = O(n^{\log_6 7}) + 6n\left(n^{\log_6 \frac{7}{6}} - 1\right) \\
&= \quad O(n^{\log_6 7}) + O(n^{1+\log_6 \frac{7}{6}}) = O(n^{\log_6 7}) + O(n^{1+\log_6 7 - \log_6 6}) = O(n^{\log_6 7}),
\end{aligned}
$$

avendo posto $k = \log_6 n$ e usato il fatto che

$$
a^{\log_6 n} = a^{\frac{\log_a n}{\log_a 6}} = n^{\frac{1}{\log_a 6}} = n^{\log_6 a}.
$$

Visto che $\log_6 7 < 2$, il secondo algoritmo è più veloce del primo.