Soluzione della seconda esercitazione di Algoritmi 1

Beniamino Accattoli

10 ottobre 2007

1 Strumenti per l'esercitazione

Definizione: $g(n) \in O(f(n))$ se esistono c e n_0 costanti positive tali che per ogni $n > n_0$ vale $g(n) \leq c \cdot f(n)$.

Somma dei primi k termini di una serie geometrica: (per $x \neq 1$)

$$\sum_{i=0}^{k} x^i = \frac{1 - x^{k+1}}{1 - x}$$

Formule di potenze e logaritmi:

$$(x^a)^b = x^{a \cdot b} = \left(x^b\right)^a \quad \log_a x = \frac{1}{\log_x a} \quad \log_a n = \frac{\log_b n}{\log_b a} \quad \log_{t^j} n = \frac{\log_t n}{j}$$

2 Esercizi

Si consideri il seguente algoritmo (si assume che il primo elemento dell'array A abbia indice 1):

algoritmo JustForFun(*array A di interi*, *intero n*) $\rightarrow$ *intero*
if $(n \leq 1)$ OR $(n > dim(A))$ **then** **return** 1
else $somma \leftarrow 0$
  for $j = 1$ **to** n **do** $somma \leftarrow somma + A[j]$
  for $j = 1$ **to** $\lfloor \log_2(n) \rfloor$ **do** $A[j] \leftarrow A[j] * 3$
  return $somma + 2 * JustForFun\,(A, \lfloor n/2 \rfloor)$

1. Calcolare JustForFun([1,2,3,4], 4).
   Risposta: 32.

2. Sia $T(A, n)$ il tempo di esecuzione dell'algoritmo JustForFun su input A e n, dove $n \leq dim(A)$. Scrivere una ricorrenza per $T(A, n)$.
   Risposta:

$$T(A, n) = \begin{cases} T(A, \lfloor n/2 \rfloor) + c_1 \cdot n & n > 1 \\ c_2 & n \leq 1 \end{cases}$$

3. Sotto quale ipotesi sull'input le parti intere ai punti 4 e 5 dell'algoritmo sono superflue? Perché, al fine dell'analisi, la parte intera al punto 4 è ininfluente?
   Risposta: L'ipotesi sull'input è che sia una potenza di due, ovvero che esista j tale che $n = 2^j$. La parte intera al punto 4 è ininfluente perché il termine logaritmico dato da quella riga è assorbito dal termine lineare '$c_1 n$'.

4. Si supponga che n sia tale che $dim(A) \geq n > 1$ e nell'ipotesi trovata al punto precedente. Applicare la definizione di T alla prima chiamata ricorsiva in $T(A, n)$. Iterare e cercare di esprimere la k-esima iterazione in modo compatto.
   Risposta:
$$T(A,n) = T(A, \frac{n}{2}) + c_1 \cdot n = T(A, \frac{n}{2^2}) + c_1 \cdot n + \frac{c_1 \cdot n}{2} =$$
$$= T(A, \frac{n}{2^3}) + \frac{c_1 \cdot n}{2^0} + \frac{c_1 \cdot n}{2^1} + \frac{c_1 \cdot n}{2^2} = .. = T(A, \frac{n}{2^k}) + c_1 \cdot n \sum_{i=0}^{k-1} \frac{1}{2^i}$$

5. Analizzando la formula trovata al punto precedente, calcolare il numero k di applicazioni della definizione di T perché $T(A, n)$ richiami $T(A, 1)$, se $n > 1$
   Risposta: Dev'essere $\frac{n}{2^k} \leq 1$, vero se e solo se $n \leq 2^k$, e passando al logaritmo in base 2 si ha $k \geq \log_2 n$

6. Calcolare l'andamento asintotico di $T(A, n)$
   Risposta: Usando la formula per la k-esima iterazione trovata prima si ha (con $k = \log_2 n$)
$$T(A,n) = T(A,1) + c_1 \cdot n \sum_{i=0}^{\log_2 n - 1} \frac{1}{2^i} = c_2 + c_1 \cdot n \cdot \frac{1 - \frac{1}{2^{log_2 n}}}{1 - \frac{1}{2}} =$$
$$= c_2 + 2 \cdot c_1 \cdot n \cdot (1 - \frac{1}{n}) \leq 2 \cdot c_1 \cdot n + c_2$$
   e $(2 \cdot c_1 \cdot n + c_2) = O(n)$. Un modo più veloce di mostrare la cosa è notando che $\sum_{i=0}^{\log_2 n - 1} \frac{1}{2^i} \leq \sum_{i=0}^{+\infty} \frac{1}{2^i}$ la quale è una serie geometrica di ragione minore di 1, e quindi convergente ad una costante h (nel nostro caso $h = 2$). E quindi si ha direttamente
$$T(A,n) = T(A,1) + c_1 \cdot n \sum_{i=0}^{\log_2 n - 1} \frac{1}{2^i} \leq 2 \cdot c_1 \cdot n + c_2$$
   La formula per trovare h è
$$\sum_{i=0}^{+\infty} x^i = \frac{1}{1-x}$$

7. Iniziamo ora a variare alcuni parametri dell'algoritmo, al fine di capire come varia la ricorrenza associata e la complessità del problema. Sostituire l'ultima riga dell'algoritmo con:

$$\textbf{return } somma + 2 * JustForFun\,(A, \lfloor n/3 \rfloor)$$

Ripetere i punti 2, 3, 4, 5 e 6. Sostituire '3' con un intero positivo u qualsiasi, diverso da 1, e ripetere l'analisi.
Risposta: si ha

$$T(A,n) = T(A,\frac{n}{3}) + c_1 \cdot n = T(A,\frac{n}{3^2}) + c_1 \cdot n + \frac{c_1 \cdot n}{3} =$$

$$= T(A,\frac{n}{3^3}) + \frac{c_1 \cdot n}{3^0} + \frac{c_1 \cdot n}{3^1} + \frac{c_1 \cdot n}{3^2} = .. = T(A,\frac{n}{3^k}) + c_1 \cdot n \sum_{i=0}^{k-1} \frac{1}{3^i}$$

. Perché sia $\frac{n}{3^k} \leq 1$ dev'essere $k \geq \log_3 n$. Per $k \geq \log_3 n$ si ha

$$T(A,n) = T(A,1) + c_1 \cdot n \sum_{i=0}^{\log_3 n - 1} \frac{1}{3^i} \leq h \cdot c_1 \cdot n + c_2$$

In particolare $h = 3/2$. E anche in questo caso si ha un $O(n)$. Si vede facilmente che qualunque sia u positivo e diverso da 1 si ottiene la stessa complessità. Infatti

$$T(A,n) = T(A,\frac{n}{u}) + c_1 \cdot n = T(A,\frac{n}{u^2}) + c_1 \cdot n + \frac{c_1 \cdot n}{u} =$$

$$= T(A,\frac{n}{u^3}) + \frac{c_1 \cdot n}{u^0} + \frac{c_1 \cdot n}{u^1} + \frac{c_1 \cdot n}{u^2} = .. = T(A,\frac{n}{u^k}) + c_1 \cdot n \sum_{i=0}^{k-1} \frac{1}{u^i}$$

. Perché sia $\frac{n}{u^k} \leq 1$ dev'essere $k \geq \log_u n$, e quindi si ha

$$T(A,n) = T(A,1) + c_1 \cdot n \sum_{i=0}^{\log_u n - 1} \frac{1}{u^i} \leq h \cdot c_1 \cdot n + c_2$$

con $h = \frac{u}{u-1}$

8. Sostituire l'ultima riga dell'algoritmo con:

$$\textbf{return } somma + JustForFun\,(A, \lfloor n/2 \rfloor) + JustForFun\,(A, \lfloor n/2 \rfloor)$$

E ripetere l'analisi.
Risposta: la ricorrenza diventa

$$T(A,n) = \begin{cases} 2 \cdot T\,(A, \lfloor n/2 \rfloor) + c_1 \cdot n & n > 1 \\ c_2 & n \leq 1 \end{cases}$$

Supponendo che n sia una potenza di 2 si può dimenticare la parte intera e scrivere

$$T(A,n) = 2 \cdot T(A, \frac{n}{2}) + c_1 \cdot n = 2(2 \cdot T(A, \frac{n}{2^2}) + \frac{c_1 \cdot n}{2}) + c_1 \cdot n =$$

$$= 2^2 \cdot T(A, \frac{n}{2^2}) + 2 \cdot c_1 \cdot n = 2^3 \cdot T(A, \frac{n}{2^3}) + 3 \cdot c_1 \cdot n = .. = 2^k T(A, \frac{n}{2^k}) + k \cdot c_1 \cdot n$$

per ricadere nel caso base dev'essere $k \geq \log_2 n$. Sostituendo $k = \log_2 n$ si trova $T(A,n) = 2^{\log_2 n} + \log_2 n \cdot c_1 \cdot n = n + c_1 n \cdot \log_2 n$, ovvero un $O(n \log n)$

9. Sostituire l'ultima riga dell'algoritmo con:

   return *somma*$+$ *JustForFun* $(A, \lfloor n/2 \rfloor) +$ *JustForFun* $(A, \lfloor n/2 \rfloor) +$ *JustForFun* $(A, \lfloor n/2 \rfloor) +$ *JustForFun* $(A, \lfloor n/2 \rfloor)$

   e ripetere l'analisi.
   Risposta: la ricorrenza diventa

$$T(A,n) = \begin{cases} 4 \cdot T(A, \lfloor n/2 \rfloor) + c_1 \cdot n & n > 1 \\ c_2 & n \leq 1 \end{cases}$$

   Supponendo che n sia una potenza di 2 si può dimenticare la parte intera e scrivere

$$T(A,n) = 4 \cdot T(A, \frac{n}{2}) + c_1 \cdot n = 4(4 \cdot T(A, \frac{n}{2^2}) + \frac{c_1 \cdot n}{2}) + c_1 \cdot n =$$

$$= 4^2 \cdot T(A, \frac{n}{2^2}) + 2 \cdot c_1 \cdot n + c_1 \cdot n = 4^3 \cdot T(A, \frac{n}{2^3}) + 4 \cdot c_1 \cdot n + 2 \cdot c_1 \cdot n + c_1 \cdot n =$$

$$= 4^3 \cdot T(A, \frac{n}{2^3}) + 2^2 \cdot c_1 \cdot n + 2^1 \cdot c_1 \cdot n + 2^0 \cdot c_1 \cdot n = .. = 4^k \cdot T(A, \frac{n}{2^k}) + c_1 \cdot n \sum_{i=0}^{k-1} 2^i$$

   per ricadere nel caso base dev'essere $k \geq \log_2 n$. Sostituendo $k = \log_2 n$ si trova $T(A,n) = 4^{\log_2 n} + c_1 \cdot n \cdot \frac{1-n}{1-2} = n^2 + c_1 \cdot n \cdot (n-1)$, ovvero un $O(n^2)$

10. Cerchiamo di ottenere una regola generale. Supponiamo che l'ultima riga dell'algoritmo sia della forma:

    return *somma*$+ \underbrace{\mathit{JustForFun}\,(A, \lfloor n/u \rfloor) + .. + \mathit{JustForFun}\,(A, \lfloor n/u \rfloor)}_{t \text{ volte}}$

    dove t e u sono due interi positivi, e $u > 1$. Ripetere l'analisi fino al punto 5. Supporre $t = u$ e risolvere la ricorrenza
    Risposta: la ricorrenza diventa

$$T(A,n) = \begin{cases} t \cdot T(A, \lfloor n/u \rfloor) + c_1 \cdot n & n > 1 \\ c_2 & n \leq 1 \end{cases}$$

Supponendo che n sia una potenza di u si può dimenticare la parte intera e scrivere

$$T(A,n) = t \cdot T(A, \frac{n}{u}) + c_1 \cdot n = t(t \cdot T(A, \frac{n}{u^2}) + \frac{c_1 \cdot n}{u}) + c_1 \cdot n =$$

$$= t^2 \cdot T(A, \frac{n}{u^2}) + \frac{t}{u} \cdot c_1 \cdot n + c_1 \cdot n = t^3 \cdot T(A, \frac{n}{u^3}) + \left(\frac{t}{u}\right)^2 \cdot c_1 \cdot n + \frac{t}{u} \cdot c_1 \cdot n + c_1 \cdot n =$$

$$= .. = t^k \cdot T(A, \frac{n}{u^k}) + c_1 \cdot n \sum_{i=0}^{k-1} \left(\frac{t}{u}\right)^i$$

Dev'essere $\frac{n}{u^k} \leq 1$, vero se e solo se $n \leq u^k$, e passando al logaritmo in base u si ha $k \geq \log_u n$. Sostituendo $k = \log_u n$ si ottiene

$$t^{\log_u n} \cdot c_2 + c_1 \cdot n \sum_{i=0}^{\log_u n - 1} \left(\frac{t}{u}\right)^i$$

Se $t = u$ si ha

$$T(A,n) = t^{\log_t n} \cdot c_2 + c_1 \cdot n \sum_{i=0}^{\log_t n - 1} 1 = n \cdot c_2 + c_1 \cdot n \cdot \log_t n$$

Ovvero $T(A,n) = O(n \cdot \log n)$

11. Continuando il punto precedente: $t = u^2$ e risolvere la ricorrenza. Supporre $t = u^j$ con j intero, e risolvere la ricorrenza.
Risposta: se $t = u^2$ la formula diventa

$$T(A,n) = \left(u^2\right)_2^{\log_u n} + c_1 \cdot n \sum_{i=0}^{\log_u n - 1} \left(\frac{u^2}{u}\right)^i = \left(u^k\right)^2 \cdot c_2 + c_1 \cdot n \sum_{i=0}^{\log_u n - 1} u^i =$$

$$= \left(u^{\log_u n}\right)^2 \cdot c_2 + c_1 \cdot n \frac{1 - u^{\log_u n}}{1 - u} =$$

$$= n^2 \cdot c_2 + c_1 \cdot n \cdot \frac{1-n}{1-u} = n^2 \cdot c_2 + \frac{c_1}{u-1} \cdot n \cdot (n-1)$$

ovvero un $O\left(n^2\right)$ (ci si ricordi che c_1, c_2 e u sono costanti).

Se $t = u^j$ si ha

$$\left(u^{\log_u n}\right)^j \cdot c_2 + c_1 \cdot n \sum_{i=0}^{\log_u n - 1} \left(u^{j-1}\right)^i$$

e ragionando come prima si trova

$$T(A,n) = n^j \cdot c_2 + \frac{c_1}{u^{j-1} - 1} \cdot n \cdot \left(n^{j-1} - 1\right)$$

ovvero un $O\left(n^j\right)$

12. Continuando: Supporre $u = t^2$ e risolvere la ricorrenza. Supporre $u = t^j$ con j intero, e risolvere la ricorrenza. (usare le formule dei logaritmi)
Risposta: se $u = t^2$ la formula diventa

$$T(A,n) = t^{\log_{t^2} n} \cdot c_2 + c_1 \cdot n \sum_{i=0}^{\log_u n - 1} \left(\frac{t}{t^2}\right)^i = t^{\frac{\log_t n}{2}} \cdot c_2 + c_1 \cdot n \sum_{i=0}^{\log_u n - 1} \left(\frac{1}{t}\right)^i =$$

$$= \left(t^{\log_t n}\right)^{1/2} \cdot c_2 + c_1 \cdot n \sum_{i=0}^{\log_u n - 1} \left(\frac{1}{t}\right)^i = n^{1/2} \cdot c_2 + c_1 \cdot n \sum_{i=0}^{\log_u n - 1} \left(\frac{1}{t}\right)^i \leq$$

$$\leq n^{1/2} \cdot c_2 + c_1 \cdot h \cdot n$$

per $h = \frac{t}{t-1}$ e quindi $T(A,n) = O(n)$. Per $u = t^j$ si ottiene

$$T(A,n) = t^{\log_{t^j} n} \cdot c_2 + c_1 \cdot n \sum_{i=0}^{\log_u n - 1} \left(\frac{t}{t^2}\right)^i = .. = n^{1/j} \cdot c_2 + c_1 \cdot n \sum_{i=0}^{\log_u n - 1} \left(\frac{1}{t}\right)^i \leq$$

$$\leq n^{1/j} \cdot c_2 + c_1 \cdot h \cdot n$$

e di nuovo $T(A,n) = O(n)$.

13. Cosa si può dire se si assume semplicemente che $u > t$? (usare le formule dei logaritmi). E se si assume $u < t$?
Risposta: Per la formula del cambiamento di base si ha $\log_u n = \frac{\log_t n}{\log_t u}$, e così possiamo scrivere

$$t^{\log_u n} \cdot c_2 + c_1 \cdot n \sum_{i=0}^{\log_u n - 1} \left(\frac{t}{u}\right)^i = \left(t^{\log_t n}\right)^{\frac{1}{\log_t u}} \cdot c_2 + c_1 \cdot n \sum_{i=0}^{\log_u n - 1} \left(\frac{t}{u}\right)^i$$

usando $\log_u t = \frac{1}{\log_t u}$ si trova

$$= n^{\log_u t} \cdot c_2 + c_1 \cdot n \sum_{i=0}^{\log_u n - 1} \left(\frac{t}{u}\right)^i$$

Se $t < u$ allora $\log_u t < 1$ e la sommatoria è maggiorabile con una costante h. Si ha quindi $T(A,n) = n^{\log_u t} \cdot c_2 + c_1 \cdot h \cdot n$, cioè $T(A,n) = O(n)$. Se invece vale $t > u$ la sommatoria vale

$$\frac{1 - \left(\frac{t}{u}\right)^{\log_u n}}{1 - \frac{t}{u}} = \frac{1 - \frac{t^{\log_u n}}{u^{\log_u n}}}{1 - \frac{t}{u}} = \frac{1 - \frac{t^{\frac{\log_t n}{\log_t u}}}{n}}{1 - \frac{t}{u}} = \frac{1 - \frac{n^{\log_u t}}{n}}{1 - \frac{t}{u}}$$

si trova quindi

$$T(A,n) = n^{\log_u t} \cdot c_2 + c_1 \cdot n \cdot \frac{1 - \frac{n^{\log_u t}}{n}}{1 - \frac{t}{u}} = n^{\log_u t} \cdot c_2 + c_1 \cdot \frac{n - n^{\log_u t}}{1 - \frac{t}{u}} =$$

$$= n^{\log_u t} \cdot c_2 + c_1 \cdot \frac{n^{\log_u t} - n}{\frac{t}{u} - 1}$$

ovvero $T(A,n) = O\left(n^{\log_u t}\right)$, dove $\log_u t > 1$ perché $t > u$.

14. Si sono viste diverse forme di ricorrenza in base ai valori dei parametri t ed u: riassumere il tutto.
Risposta:

$$T(A,n) = \begin{cases} O\left(n^{\log_u t}\right) & t > u \\ O(n \log n) & t = u \\ O(n) & t < u \end{cases}$$

15. Per chi desiderasse un esercizio simile ma diverso [quest'esercizio e il seguente non sono stati svolti durante l'esercitazione]: ripetere tutta l'esercitazione eliminando la linea 3 dell'algoritmo.

16. Per addomesticare i logaritmi: dimostrare le formule date per l'esercitazione (per la più difficile trovate dei suggerimenti nella prima esercitazione).
Risposta: la prima è ovvia. Per la seconda si usano $x = a^{\log_a x}$ e $a = x^{\log_x a}$:

$$a = x^{\log_x a} = \left(a^{\log_a x}\right)^{\log_x a} = a^{\log_a x \cdot \log_x a}$$

quindi $1 = \log_a x \cdot \log_x a$ da cui $\log_a x = \frac{1}{\log_x a}$
Per la terza, si rimanda alla prima esercitazione.
L'ultima: si ha $\left(t^j\right)^{\log_{t^j} n} = n = t^{\log_t n}$. Ovvero $\left(t^{j \cdot \log_{t^j} n}\right) = t^{\log_t n}$, da cui $\log_{t^j} n = \frac{\log_t n}{j}$