3

Ricorrenze

Nel caso di algoritmi ricorsivi (ad esempio, merge sort, ricerca binaria, ricerca del massimo e/o del minimo), il tempo di esecuzione può essere descritto da una funzione ricorsiva, ovvero da un'equazione che descrive una funzione in termini del suo valore con input più piccoli. Esempio

$$T(n) = \begin{cases} 2T(n/2) + n & \text{se } n > 1 \\ 1 & \text{se } n = 1 \end{cases}$$

3.1 Metodo iterativo

Il metodo iterativo consiste nello srotolare la ricorrenza, cioè "richiamare", la ricorrenza un certo numero di volte (ogni volta con un argomento di dimensione più piccola) fino quando non è possibile esprimerla come una somma di termini dipendenti solo da n e dalle condizioni iniziali. Questo metodo, così come la sua variante – il *metodo dell'albero di ricorsione* – si rivela particolarmente utile quando non si hanno idee precise sull'ordine di grandezza della ricorrenza. Entrambi i metodi permettono infatti di "capire come si dipanano i costi nella ricorsione".

Exercise 3.1. Si risolva le seguenti ricorrenze usando il metodo iterativo; assumete che, in ogni caso, $T(1) = 1$.

1. $T(n) = c + T(n/2)$
2. $T(n) = 1 + 2T(n/2)$
3. $T(n) = n^3 + 3T(n/3)$

Soluzione. 1. Per la prima ricorrenza:

$$\begin{aligned} T(n) &= c + T(n/2) \\ &= c + c + T(n/4) = && \text{dopo 2 passi} \\ &= c + c + c + T(n/8) && \text{dopo 3 passi} \\ &= c + c + c + c + T(n/16) && \text{dopo 4 passi} \\ &= \dots \\ &= kc + T(n/2^k) && \text{dopo } k \text{ passi} \end{aligned}$$

Continuiamo a srotolare la ricorrenza fin quando $n/2^k = 1$ ossia fin quando $k = \log n$. Allora, possiamo riscrivere $T(n)$ come segue:

$$T(n) = c \log n + T(1) = c \log n + 1 = \Theta(\log n)$$

2.

$$\begin{aligned}
T(n) &= 1 + 2T(n/2) \\
&= 1 + 2\big(1 + 2T(n/4)\big) \\
&= 1 + 2 + 4T(n/4) && \text{dopo 2 passi} \\
&= 1 + 2 + 4\big(1 + 2T(n/8)\big) \\
&= 1 + 2 + 4 + 8T(n/8) && \text{dopo 3 passi} \\
&= 1 + 2 + 4 + 8\big(1 + 2T(n/16)\big) \\
&= 1 + 2 + 4 + 8 + 16T(n/16) && \text{dopo 4 passi} \\
&= \dots \\
&= 1 + 2 + 4 + \dots 2^{k-1} + 2^k T(n/2^k) && \text{dopo } k \text{ passi} \\
&= \sum_{i=0}^{k-1} 2^i + 2^k \cdot T(n/2^k)
\end{aligned}$$

Di nuovo, ci fermiamo quando $n/2^k = 1$, ossia fin quando $k = \log n$. Allora, possiamo riscrivere $T(n)$ come segue:

$$T(n) = \sum_{i=0}^{\log n - 1} 2^i + 2^{\log n} \cdot T(1) = \sum_{i=0}^{\log n - 1} 2^i + n = \frac{1 - 2^{\log n}}{1 - 2} + n =$$

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

3.

$$\begin{aligned}
T(n) &= n^3 + 3T(n/3) \\
&= n^3 + 3\big((n/3)^3 + 3T(n/9)\big) \\
&= n^3 + n^3/9 + 9T(n/9) && \text{dopo 2 passi} \\
&= n^3 + n^3/9 + 9\big((n/9)^3 + 3T(n/27)\big) = \\
&= n^3 + n^3/9 + n^3/9^2 + 27T(n/27) && \text{dopo 3 passi} \\
&= n^3 + n^3/9 + n^3/9^2 + 27\big((n/27)^3 + 3T(n/81)\big) \\
&= n^3 + n^3/9 + n^3/9^2 + n^3/27^2 + 81T(n/81) \\
&= n^3 + n^3/9 + n^3/9^2 + n^3/9^3 + 81T(n/81) && \text{dopo 4 passi} \\
&= \dots \\
&= \sum_{i=0}^{k-1} n^3/9^i + 3^k T(n/3^k) \\
&= n^3 \sum_{i=0}^{k-1} (1/9)^i + 3^k T(n/3^k) && \text{dopo } k \text{ passi}
\end{aligned}$$

In questo caso ci fermiamo quando $n/3^k = 1$ e quindi quando $k = \log_3 n$

$$T(n) = n^3 \sum_{i=0}^{\log_3 n - 1} (\frac{1}{9})^i + n = n^3 \, \frac{1 - (\frac{1}{9})^{\log_3 n}}{1 - \frac{1}{9}} + n = n^3 \, \frac{1 - (\frac{1}{9})^{\log_3 n}}{\frac{8}{9}} + n$$

$$= \frac{8}{9}n^3\Big(1 - \frac{1}{9^{\log_3 n}}\Big) + n$$

Poichè $9^{\log_3 n} = n^{\log_3 9} = n^2$ (poichè $a^{\log_b c} = c^{\log_b a}$):

$$T(n) = \frac{9}{8}n^3\Big(1 - \frac{1}{n^2}\Big) + n = \frac{9}{8}n^3 - \frac{9}{8}n + n = \frac{9}{8}n^3 - \frac{1}{8}n = \Theta(n^3)$$

3.2 Metodo dell'albero di ricorsione

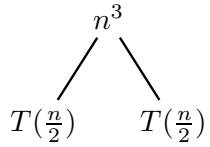
È una variante del metodo iterativo. Un albero di ricorsione permette di visualizzare lo sviluppo della ricorsione e dei costi; ogni nodo di questo albero rappresenta il costo di un particolare sottoproblema appartenente all'insieme delle chiamate ricorsive della funzione. Il valore della ricorrenza viene calcolato sommando prima i costi dei nodi su ciascun livello, e poi determinando il costo totale di tutti i livelli dell'albero di ricorsione.

Exercise 3.2. Risolvere con il metodo dell'albero di ricorsione la seguente ricorrenza

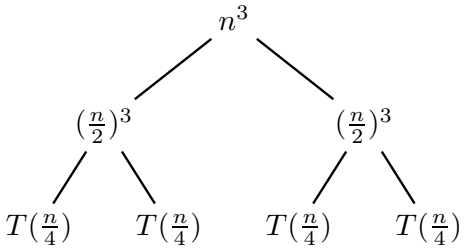
$$T(n) = \begin{cases} n^3 + 2T(n/2) & \text{se } n > 1 \\ 1 & \text{se } n = 1 \end{cases}$$

Soluzione. **Costruzione dell'albero di ricorsione**:

- inizialmente, l'albero ha un solo nodo (la radice) che rappresenta il costo complessivo della chiamata di $T(n)$
- la radice viene espansa secondo la definizione della ricorrenza. Nel nostro caso otteniamo il seguente albero



  ossia, una rappresentazione grafica del fatto che il costo di una chiamata di $T(n)$ è pari n^3 più il costo di due chiamate di $T(n/2)$
- in maniera simile, espandiamo i nodi relativi alle due chiamate di $T(n/2)$ ottendo



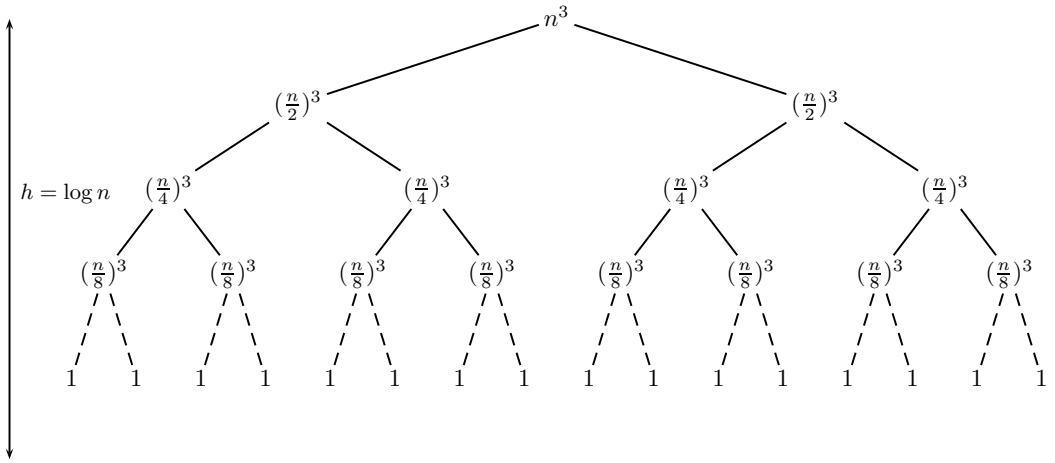


Fig. 3.1. Albero di ricorsione per $T(n) = 2T(n/2) + n^3$

- a questo punto, ognuno di quattro nodi relativi ad una chiamata di $T(n/4)$ genererà un albero la cui radice (etichettata con $(n/4)^3$) ha due figli che rappresentano il costo di una chiamata di $T(n/8)$.
- procediamo in questo modo fino a raggiungere la condizione di contorno, ossia quell particolare valore di n (tipicamente 1) che non da luogo ad ulteriori chiamate ricorsive. Otteniamo l'albero descritto in Figura 3.1. Si noti che l'ultimo livello corrisponde ad una chiamata di $T(n/2^h)$ con $\frac{n}{2^h} = 1$ e quindi $h = \log n$ (qui h è l'altezza dell'albero).

Calcolo dei costi associati all'albero

Ogni livello i dell'albero (con $i = 0, \ldots, \log n$) ha 2^i nodi ognuno dei quali ha un costo di $(\frac{n}{2^i})^3$. Il costo di ciascun livello i è pari a:

$$c_i = 2^i \cdot (\frac{n}{2^i})^3 = \frac{n^3}{(2^i)^2} = \frac{n^3}{4^i}$$

$T(n)$ è pari alla somma della somma dei costi associati ai nodi dell'albero di ricorsione, in particolare

$$T(n) = \sum_{i=0}^{h} c_i = \sum_{i=0}^{\log n} \frac{n^3}{4^i} = n^3 \cdot \sum_{i=0}^{\log n} (\frac{1}{4})^i = n^3 \Big(\frac{1 - (\frac{1}{4})^{\log n + 1}}{1 - \frac{1}{4}} \Big) =$$

$$n^3 \Big(\frac{1 - \frac{1}{4} \cdot (\frac{1}{4})^{\log n}}{\frac{3}{4}} \Big) = \frac{4}{3} n^3 \Big(1 - \frac{1}{4} \cdot \Big(\frac{1}{4} \Big)^{\log n} \Big) =$$

$$\frac{4}{3} n^3 \Big(1 - \frac{1}{4} \cdot \frac{1}{4^{\log n}} \Big) = \frac{4}{3} n^3 \Big(1 - \frac{1}{4} \cdot \frac{1}{n^{\log 4}} \Big) =$$

$$\frac{4}{3} n^3 \Big(1 - \frac{1}{4} \cdot \frac{1}{n^2} \Big) = \frac{4}{3} n^3 - \frac{1}{3} n = \Theta(n^3)$$

Exercise 3.3. Risolvere la seguente ricorrenza con il metodo dell'albero di ricorsione, assumendo che $T(1) = 1$

$$T(n) = T(\frac{n}{4}) + T(\frac{n}{2}) + cn$$

Soluzione. L'albero di ricorsione per $T(n)$ è illustrato in Figura 3.2. Come per ogni altro albero di ricorsione, le dimensioni dei sottoproblemi diminuiscono via via che ci si allontana dalla radice, fino a raggiungere le condizioni di contorno (la condizione di contorno è il particolare valore di n (in genere 1) che non da luogo ad ulteriori chiamate ricorsive). Il "problema" con questo tipo di ricorrenze è che le condizioni di contorno vengono raggiunte a diverse distanze dalla radice. Questo perchè il nodo più a sinistra e quello più a destra di un generico livello i corrispondono, rispettivamente, a sottoproblemi di dimensione $\frac{n}{4^i}$ e $\frac{n}{2^i}$, e $\frac{n}{4^i}$ decresce (e quindi arriverà ad assumere il valore 1) molto più rapidamente di $\frac{n}{2^i}$. In altri termini, la condizione di contorno viene raggiunta a sinistra ad una distanza h dalla radice con h tale che $\frac{n}{4^h} = 1$ (ossia $n = 4^h$ e quindi $h = \log_4 n$). A destra, invece, la condizione di contorno viene raggiunta ad una distanza k dalla radice con k tale che $\frac{n}{2^k} = 1$ (e quindi $k = \log n$).

Il costo di ogni livello i con $i = 0, \ldots, h$ è pari a $c_i = (\frac{3}{4})^i cn$. Osservate che:

- $c_0 = cn = (\frac{3}{4})^0 cn$
- $c_1 = \frac{cn}{4} + \frac{cn}{2} = (\frac{1}{4} + \frac{1}{2})cn = \frac{3}{4}cn$
- $c_2 = \frac{cn}{16} + \frac{cn}{8} + \frac{cn}{8} + \frac{cn}{4} = (\frac{1}{16} + \frac{1}{8} + \frac{1}{8} + \frac{1}{4})cn = \frac{1+2+2+4}{16}cn = \frac{9}{16}cn = (\frac{3}{4})^2 cn$
- $c_3 = (\frac{1}{64} + \frac{1}{32} + \frac{1}{32} + \frac{1}{16} + \frac{1}{32} + \frac{1}{16} + \frac{1}{16} + \frac{1}{8})cn = \frac{1+2+2+4+2+4+4+8}{64}cn = \frac{27}{64}cn = (\frac{3}{4})^3 cn$

Il costo di ciascun livello $i = h+1, ..., k$ è sicuramente minore o uguale di $(\frac{3}{4})^i cn$ (mancano alcuni nodi a sinistra e, di conseguenza, il contributo di questi nodi al costo complessivo del livello). Ricapitolando: l'albero è costituito da $\log n + 1$ livelli; inoltre, per ogni $i = 0 \ldots \log n$, il costo $c_i \leq (\frac{3}{4})^i cn$. Quindi:

$$T(n) = \sum_{i=0}^{\log n} c_i \leq \sum_{i=0}^{\log n} \left(\frac{3}{4}\right)^i cn = cn \sum_{i=0}^{\log n} \left(\frac{3}{4}\right)^i = cn \; \frac{1 - \left(\frac{3}{4}\right)^{\log n+1}}{1 - \frac{3}{4}} =$$

$$cn \; \frac{1 - \left(\frac{3}{4}\right)^{\log n+1}}{\frac{1}{4}} = 4cn \left(1 - \frac{3}{4} \cdot \left(\frac{3}{4}\right)^{\log n}\right) = 4cn \left(1 - \frac{3}{4} \cdot \left(\frac{3^{\log n}}{4^{\log n}}\right)\right) =$$

$$4cn \left(1 - \frac{3}{4} \cdot \left(\frac{n^{\log 3}}{n^{\log 4}}\right)\right) = 4cn \left(1 - \frac{3}{4} \cdot \left(\frac{n^a}{n^2}\right)\right)$$

dove $1 < a = \log 3 < 2$. Quindi:

$$T(n) \leq 4cn - 4cn\frac{3}{4}\frac{n^a}{n^2} = 4cn - 3c\frac{n^a}{n} = 4cn - 3c\, n^{a-1} = \Theta(n)$$

poichè $a - 1 < 2 - 1 = 1$. Infine, $T(n) \leq g(n) = 4cn - 3c\, n^{a-1}$ e $g(n) = \Theta(n)$ implica $T(n) = O(n)$.

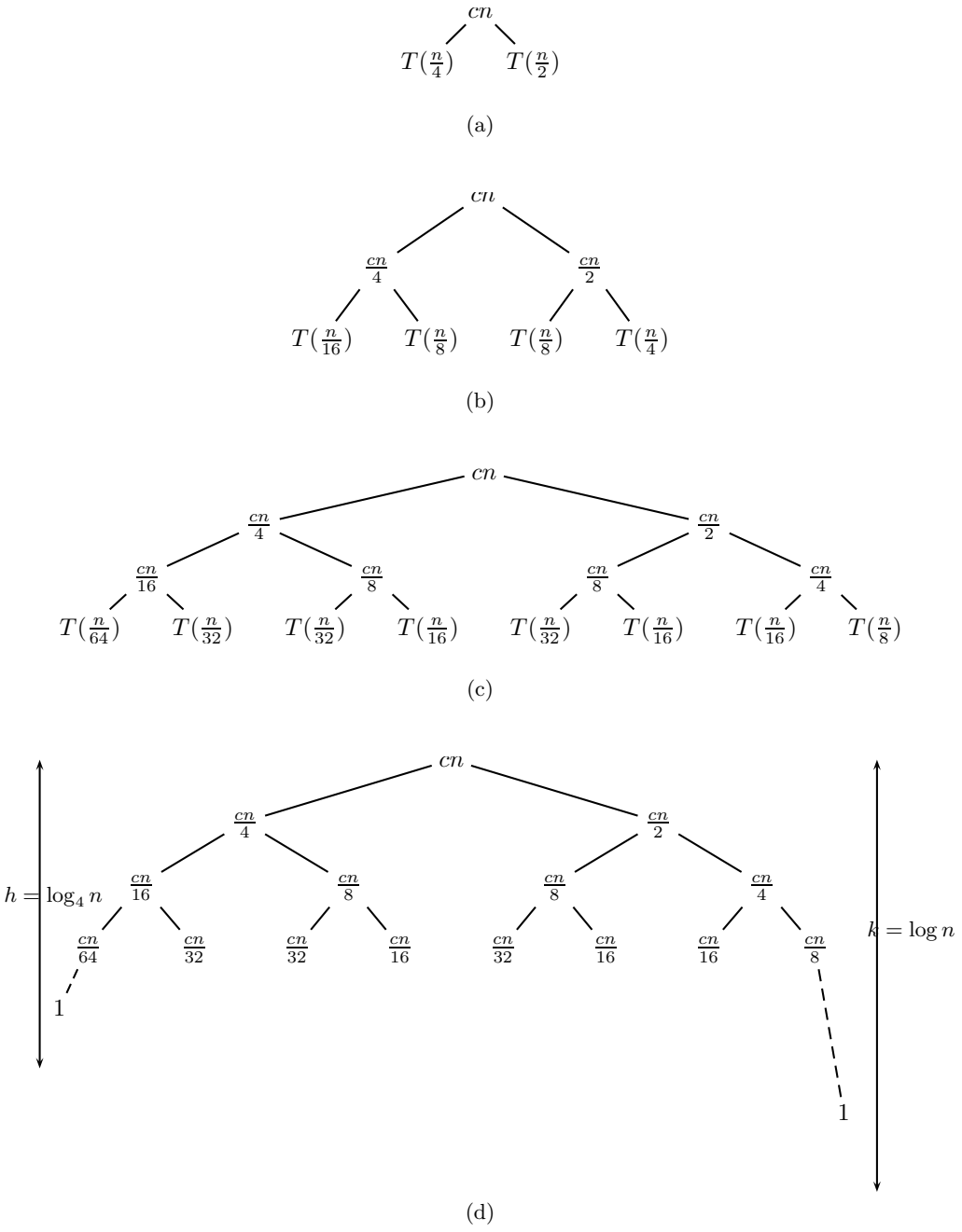


Fig. 3.2. Albero di ricorsione per $T(n) = T(n/4) + T(n/2) + cn$

Exercise 3.4. Risolvere con il metodo dell'albero di ricorsione la seguente ricorrenza

$$T(n) = \begin{cases} 3T(n/4) + \sqrt{n} \text{ se } n > 1 \\ 1 \qquad\qquad \text{se } n = 1 \end{cases}$$

Soluzione. [1]

- Ogni livello i dell'albero di ricorsione per $T(n)$ (vedi Figura 3.3) ha esattamente 3^i nodi ognuno dei quali ha un costo di $\frac{\sqrt{n}}{2^i}$. Il costo complessivo del livello i è: $c_i = 3^i \cdot \frac{\sqrt{n}}{2^i} = (\frac{3}{2})^i \sqrt{n}$
- l'ultimo livello h contiene nodi (in realtà foglie) corrispondenti al costo di una chiamata di $T(\frac{n}{4^h})$ con $\frac{n}{4^h} = 1$ e quindi $h = \log_4 n$

  Allora

$$T(n) = \sum_{i=0}^{h} (\frac{3}{2})^i \sqrt{n} = \sqrt{n} \sum_{i=0}^{\log_4 n} (\frac{3}{2})^i = \sqrt{n} \ \cdot \ \frac{1 - (\frac{3}{2})^{\log_4 n + 1}}{1 - \frac{3}{2}} =$$

$$\sqrt{n} \ \cdot \ \frac{1 - (\frac{3}{2})^{\log_4 n + 1}}{-\frac{1}{2}} = 2\sqrt{n}\Big(\Big(\frac{3}{2}\Big)^{\log_4 n + 1} - 1\Big) = 2\sqrt{n}\Big(\frac{3}{2}\Big(\frac{3}{2}\Big)^{\log_4 n} - 1\Big) =$$

$$2\sqrt{n}\Big(\frac{3}{2} \cdot \frac{3^{\log_4 n}}{2^{\log_4 n}} - 1\Big) = 2\sqrt{n}\Big(\frac{3}{2} \cdot \frac{n^{\log_4 3}}{n^{\log_4 2}} - 1\Big) = 2\sqrt{n}\Big(\frac{3}{2} \cdot \frac{n^{\log_4 3}}{n^{\frac{1}{2}}} - 1\Big) =$$

$$2\sqrt{n}\Big(\frac{3}{2} \cdot \frac{n^{\log_4 3}}{\sqrt{n}} - 1\Big) = 3n^{\log_4 3} - 2\sqrt{n}$$

Infine $\frac{1}{2} = \log_4 2 < \log_4 3$, implica $T(n) = 3n^{\log_4 3} - 2\sqrt{n} = \Theta(n^{\log_4 3})$ poichè $n^{\log_4 3}$ è il termine di ordine superiore

3.3 Medoto della sostituzione

Il metodo della sostituzione prevede due passi. Nel primo passo si stima l'ordine di grandezza asintotico per $T(n)$. Nel secondo passo si dimostra la correttezza dell'ordine di grandezza stimato. Il metodo si rivela utile quando si ha già un'idea della soluzione alla ricorrenza studiata.

Exercise 3.5. Si consideri la seguente ricorrenza

$$T(n) = \begin{cases} n^3 + 3T(n/3) \text{ se } n > 1 \\ 1 \qquad\qquad \text{se } n = 1 \end{cases}$$

Dimostrare, applicando il metodo della sostituzione, che $T(n) = O(n^3)$

[1] per la risoluzione di questo esercizio occorre ricordarsi che $\sqrt{n} = n^{\frac{1}{2}}$

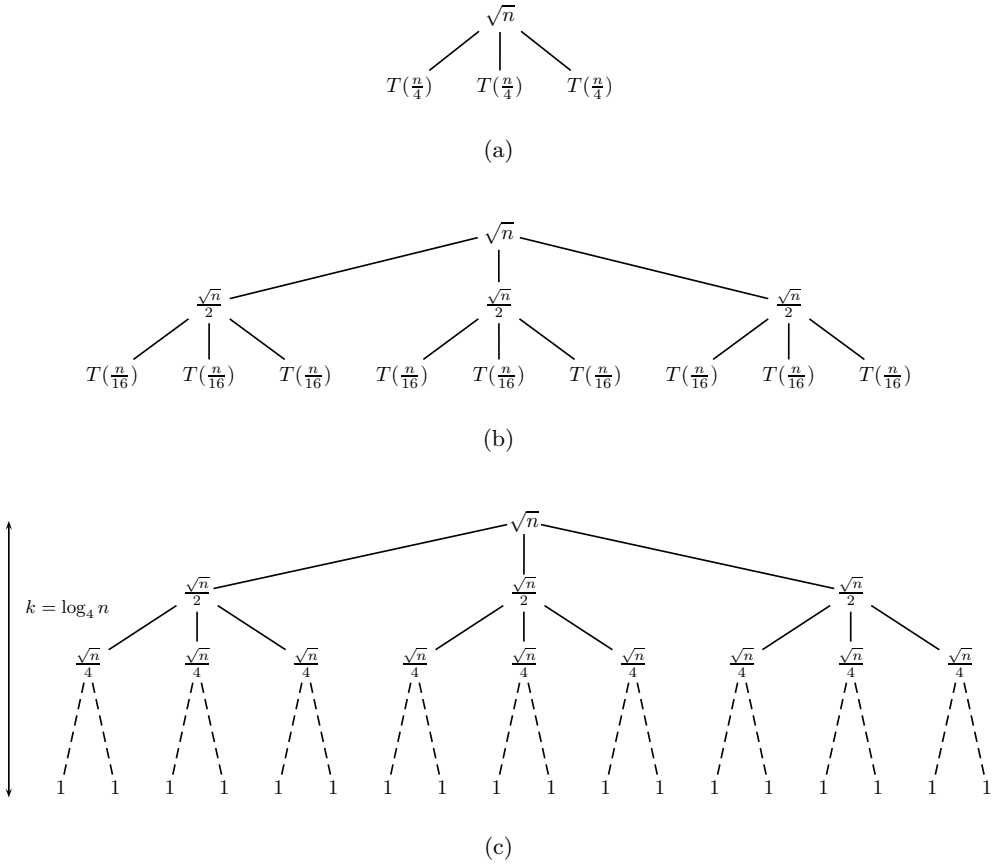


Fig. 3.3. Albero di ricorsione per $T(n) = 3T(n/4) + \sqrt{n}$

Soluzione. Dobbiamo dimostrare che esistono delle costanti positive c ed n_0 tali che $T(n) \leq cn^3$ per ogni $n \geq n_0$. Procediamo per induzione su n.

- *Caso base* $n = 1$. Se scegliamo $c \geq 1$ allora $T(1) = 1 \leq c = c1^3$.
- *Passo induttivo* $n > 1$.

$$\begin{aligned} T(n) &= n^3 + 3T(\tfrac{n}{3}) \\ &\leq n^3 + 3c(\tfrac{n}{3})^3 \quad \text{per ip. induttiva } T(\tfrac{n}{3}) \leq c(\tfrac{n}{3})^3 \\ &= n^3 + \tfrac{1}{9}cn^3 \\ &= (1 + \tfrac{1}{9}c)n^3 \\ &\leq cn^3 \end{aligned}$$

L'ultima disuguaglianza è vera solo se la costante c è scelta in maniera tale che $1 + \frac{1}{9}c \leq c$, ossia se $c - \frac{1}{9}c \geq 1$, il che implica $\frac{8}{9}c \geq 1$ e quindi $c \geq \frac{9}{8}$. Ricapitolando se scegliamo $c \geq \frac{9}{8}$ e $n_0 = 1$, allora $T(n) \leq cn^3$ per ogni $n \geq n_0$.

Exercise 3.6. Si consideri la seguente ricorrenza

$$T(n) = \begin{cases} T(n/2) + \log n & \text{se } n > 1 \\ 1 & \text{se } n = 1 \end{cases}$$

Dimostrare, applicando il metodo della sostituzione, che $T(n) = O(\log^2 n)$ dove $\log^2 n = (\log n)^2$.

Soluzione. Dobbiamo dimostrare che esistono delle costanti positive c, n_0 tali che $T(n) \leq c \log^2 n$ per ogni $n \geq n_0$.

- *Casi base* $n = 1$ $T(1) = 1 \not\leq c \log^2 1 = c \cdot 0^2 = 0$. Analiziamo il caso in cui $n = 2$. $T(2) = T(\frac{2}{2}) + \log 2 = 1 + 1 = 2 \leq c \log^2 2 = c$ a patto di scegliere $c \geq 2$.
- *Passo induttivo* $n > 2$.

$$\begin{aligned}
T(n) &= T(\tfrac{n}{2}) + \log n && \text{per ip. induttiva } T(\tfrac{n}{2}) \leq c \log^2(\tfrac{n}{2}) \\
&\leq c \log^2(\tfrac{n}{2}) + \log n && \log^2(\tfrac{n}{2}) = (\log \tfrac{n}{2})^2 = (\log n - 1)^2 \\
&= c(\log n - 1)^2 + \log n && (\log n - 1)^2 = \log^2 n - 2 \log n + 1 \\
&= c \log^2 n - 2c \log n + c + \log n \\
&= c \log^2 n - (2c - 1) \log n + c \\
&= c \log^2 n - \big((2c - 1) \log n - c\big)
\end{aligned}$$

Ora, $n \geq 2$ implica $\log n \geq 1$ e quindi $(2c-1) \log n - c \geq (2c-1) - c = c - 1 \geq 1$ poichè $c \geq 2$ (vedi il caso base). Quindi:

$$\begin{aligned}
T(n) &\leq c \log^2 n - \big((2c - 1) \log n - c\big) \\
&\leq c \log^2 n - 1 \\
&\leq c \log^2 n
\end{aligned}$$

Exercise 3.7. Si consideri la seguente ricorrenza

$$T(n) = \begin{cases} T(\frac{n}{2}) + T(\frac{n}{4}) + n & \text{se } n > 1 \\ 1 & \text{se } n = 1 \end{cases}$$

Dimostrare, applicando il metodo della sostituzione che $T(n) = O(n)$

Soluzione. Dobbiamo dimostrare che esistono delle costanti positive c ed n_0 tali che $T(n) \leq cn$ per ogni $n \geq n_0$. Procediamo per induzione su n.

- *Caso base* $n = 1$. Se scegliamo $c \geq 1$ allora $T(1) = 1 \leq c = c1$.
- *Passo induttivo* $n > 1$.

$$\begin{aligned} T(n) &= T(\tfrac{n}{2}) + T(\tfrac{n}{4}) + n \text{ per ip. induttiva } T(\tfrac{n}{2}) \leq c\tfrac{n}{2} \text{ e } T(\tfrac{n}{4}) \leq c\tfrac{n}{4} \\ &= c\tfrac{n}{2} + c\tfrac{n}{4} + n \\ &= (\tfrac{c}{2} + \tfrac{c}{4} + 1)n \\ &= (\tfrac{3}{4}c + 1)n \\ &\leq cn \end{aligned}$$

L'ultima disuguaglianza è vera se scegliamo c in maniera tale che $\frac{3}{4}c + 1 \leq c$, quindi $c - \frac{3}{4}c = \frac{1}{4}c \geq 1$. In definitiva, basta scegliere $c \geq 4$.

Exercise 3.8. Fornire un limite asintotico stretto per la ricorrenza

$$T(n) = \begin{cases} 2T(\frac{n-3}{2}) + n \text{ se } n > 3 \\ \quad 1 \qquad\quad \text{se } 1 \leq n \leq 3 \end{cases}$$

Soluzione. La ricorrenza $T(n)$ è molto simile alla ricorrenza

$$T_1(n) = \begin{cases} 2T_1(\frac{n}{2}) + n \text{ se } n > 1 \\ \quad 1 \qquad\quad \text{se } n = 1 \end{cases}$$

che ha il seguente limite asintotico stretto $T_1(n) = \Theta(n \log n)$. Dimostriamo che $T(n) = \Theta(n \log n)$ usando il metodo della sostituzione. La prova consiste di due parti: (1) dimostriamo che $T(n) = O(n \log n)$ e poi (2) dimostriamo che $T(n) = \Omega(n \log n)$.

(1) Per definizione $T(n) = O(n \log n)$ se esistono delle costanti positive c, n_0 tali che $T(n) \leq cn \log n$ per ogni $n \geq n_0$. Procediamo per induzione su n.

Casi base

- $n = 1$: $T(1) = 1 \not\leq c1 \log 1 = 0$ (per $n = 1$ la proprietà non è verificata, questo significa che n_0 deve essere maggiore di 1).
- $n = 2$: $T(2) = 1 \leq c2 \log 2 = 2c$ se scegliamo $c \geq \frac{1}{2}$.
- $n = 3$: $T(3) = 1 \leq c3 \log 3 = 3c \log 3$; vera se $c \geq \frac{1}{2}$. Infatti, se $c \geq \frac{1}{2}$ allora $3c \log 3 \geq 3c \log 2 = 3c \geq \frac{3}{2} \geq 1$.

Passo induttivo: $n > 3$

$$\begin{aligned} T(n) &= 2T(\tfrac{n-3}{2}) + n \\ &\leq 2(c\tfrac{n-3}{2} \log(\tfrac{n-3}{2})) + n \\ &\leq 2(c\tfrac{n}{2} \log(\tfrac{n}{2})) + n \\ &\leq cn(\log n - 1) + n \\ &= cn \log n - cn + n \\ &= cn \log n - n(c-1) \quad \text{poichè } c \geq 1 \\ &\leq cn \log n \end{aligned}$$

Ricapitolando, se scegliamo la costante $c \geq 1$, $T(n) \leq cn \log n$ per ogni $n \geq n_0 = 2$

(2) Per definizione $T(n) = \Omega(n \log n)$ se esistono delle costanti **positive** c, n_0 tali che $T(n) \geq cn \log n$ per ogni $n \geq n_0$. Procediamo per induzione su n.

Casi base

- $n = 1$: $T(1) = 1 \geq c1 \log 1 = 0$ per ogni c.
- $n = 2$: $T(2) = 1 \geq c \cdot 2 \log 2 = 2c$ se scegliamo $c \leq \frac{1}{2}$.
- $n = 3$: $T(3) = 1 \geq c \cdot 3 \log 3 = 3c \log 3$. Poichè $3c \log 3 \leq 3c \log 4 = 6c$, $c \leq \frac{1}{6}$ implica $3c \log 3 \leq 6c \leq 1$. Possiamo scegliere $c \leq \frac{1}{6}$ (e quindi anche $c \leq \frac{1}{2}$).

Passo induttivo: $n > 3$

$$
\begin{aligned}
T(n) &= 2T(\tfrac{n-3}{2}) + n \\
&\geq 2(c\tfrac{n-3}{2} \log(\tfrac{n-3}{2})) + n \\
&= c(n-3) \log(\tfrac{n-3}{2}) + n && n > 3 \text{ implica } \tfrac{n-3}{2} \geq \tfrac{n}{8} \\
&\geq c(n-3) \log(\tfrac{n}{8}) + n \\
&= c(n-3)(\log n - 3) + n \\
&= c\big(n \log n - 3n - 3 \log n + 9\big) + n \\
&= cn \log n - 3cn - 3c \log n + 9c + n \\
&\geq cn \log n - 3cn - 3c \log n + n \\
&= cn \log n + (1 - 3c)n - 3c \log n && \text{se } c \leq \tfrac{1}{6}, 1 - 3c \geq 1 - 3 \cdot \tfrac{1}{6} = \tfrac{1}{2} \\
&\geq cn \log n + \tfrac{1}{2}n - 3c \log n && \text{se } c \leq \tfrac{1}{6}, 3c \leq 3 \cdot \tfrac{1}{6} = \tfrac{1}{2} \\
&\geq cn \log n + \tfrac{1}{2}n - \tfrac{1}{2} \log n \\
&\geq cn \log n + \tfrac{1}{2}(n - \log n) \\
&\geq cn \log n
\end{aligned}
$$

Ricapitolando, se scegliamo la costante positiva $c \leq 1/6$, $T(n) \geq cn \log n$ per ogni $n \geq n_0 = 1$

3.4 Metodo dell'esperto

Exercise 3.9. Applicare il metodo dell'esperto per determinare i limiti asintotici stretti per le seguenti ricorrenze (assumete $T(n) = 1$ per $n = 1$)

1. $T(n) = 4T(n/2) + n$
2. $T(n) = 4T(n/2) + n^2$
3. $T(n) = 4T(n/2) + n^3$

Soluzione. In tutti i casi abbiamo $a = 4$, $b = 2$ e $\log_b a = 2$. Cambia invece il "rapporto" tra $f(n)$ e $n^{\log_b a} = n^2$. Come vedremo, le tre ricorrenze corrispondono ad un diverso con un caso del teorema del master.

1. $f(n) = n$ ed $f(n) = O(n) = O(n^{\log_b a - \varepsilon})$ con $\varepsilon = 1 > 0$; in altri termini, $f(n)$ ha $n^{\log_b a}$ come un limite superiore. Siamo, quindi, nel primo caso del teorema del master e $T(n) = \Theta(n^{\log_b a}) = \Theta(n^2)$.

2. $f(n) = n^2$ ed $f(n) = \Theta(n^2) = \Theta(n^{\log_b a})$ (ossia, $n^{\log_b a}$ è un limite stretto per $f(n)$). Secondo caso del teorema del master: $T(n) = \Theta(n^{\log_b a} \log n) = \Theta(n^2 \log n)$
3. $f(n) = n^3$ ed $f(n) = \Omega(n^3) = \Omega(n^{\log_b a + \varepsilon})$ con $\varepsilon = 1 > 0$ (ossia, $n^{\log_b a}$ è un limite inferiore per $f(n)$). Inoltre, $af(\frac{n}{b}) = 4(\frac{n}{2})^3 = 4\frac{n^3}{8} = \frac{1}{2}n^3 \leq cn^3$ (se scegliamo $c \leq \frac{1}{2} < 1$). Terzo caso del teorema del master: $T(n) = \Theta(f(n)) = \Theta(n^3)$

Exercise 3.10. La ricorrenza $T(n) = 8T(\frac{n}{2}) + n^2$ descrive il tempo di esecuzione di un algoritmo A. Un altro algoritmo A' ha un tempo di esecuzione $T'(n) = aT'(\frac{n}{4}) + n^2$ dove a è una costante intera. Quale è il più grande valore di a che rende A' asintoticamente più veloce di A?

Soluzione. Risolviamo innanzitutto la prima ricorrenza. In questo caso $a = 8$, $b = 2$ e $\log_b a = \log 8 = 3$. Allora $f(n) = n^2 = O(n^2) = O(n^{\log_b a - 1})$ (caso 1 del teorema del master) e $T(n) = \Theta(n^{\log_b a}) = \Theta(n^3)$. Risolviamo ora la seconda ricorrenza distinguendo in base ai seguenti valori della costante a.

$a < 16$. In questo caso, $\log_b a = \log_4 a < \log_4 16 = 2$ e $f(n) = n^2 = \Omega(n^2) = \Omega(n^{\log_4 a + \epsilon})$ con $\epsilon = 2 - \log_4 a > 0$. Inoltre, $af(\frac{n}{b}) = af(\frac{n}{4}) = a(\frac{n}{4})^2 = \frac{a}{16}n^2 = \frac{a}{16}f(n)$ con $\frac{a}{16} < 1$ (terzo caso del teorema del master). Allora, $T'(n) = \Theta(f(n)) = \Theta(n^2)$ e, quindi, A' è asintoticamente più veloce di A.

$a = 16$. In questo caso, $\log_b a = \log_4 16 = 2$ e $f(n) = n^2 = \Theta(n^2) = \Theta(n^{\log_b a})$ (caso 2 del teorema del master). Quindi $T'(n) = \Theta(n^{\log_b a} \log n) = \Theta(n^2 \log n)$; di nuovo, A' è asintoticamente più veloce di A.

$16 < a < 64$. $\log_b a > \log_4 16 = 2$ e $f(n) = n^2 = O(n^2) = O(n^{\log_b a - \epsilon})$ con $\epsilon = \log_b a - 2 > 0$ (caso 1 del teorema del master). Quindi $T'(n) = \Theta(n^{\log_b a})$. Poichè $a < 64$ implica $\log_b a < \log_4 64 = 3$ e $n^{\log_b a} < n^3$, possiamo ancora dedurre che A' è asintoticamente più veloce di A.

$a \geq 64$. In questo caso, $\log_b a \geq \log_4 64 = 3$ e $f(n) = n^2 = O(n^2) = O(n^{\log_b a - \epsilon})$ con $\epsilon = \log_b a - 2 > 0$ (di nuovo è il caso 1 del teorema del master). Quindi $T'(n) = \Theta(n^{\log_b a})$. La differenza rispetto al caso precedente è che $a \geq 64$ implica $n^{\log_b a} \geq 3$. In realtà, abbiamo $n^{\log_b a} = n^3$ (nel caso in cui $a = 64$) oppure $n^{\log_b a} = n^{3+\epsilon}$ per qualche $\epsilon > 0$ (questo nel caso in cui $a > 64$). In entrambi i casi, A' *non* è asintoticamente più veloce di A.

Questo dimostra che il più grande valore di a che rende A' asintoticamente più veloce di A è 63.

Exercise 3.11. Dire se il metodo dell'esperto può essere usato per risolvere la ricorrenza $T(n) = 4T(\frac{n}{2}) + n^2 \log n$. Motivare la risposta fornita.

Soluzione. In questo caso, $n^{\log_b a} = n^{\log 4} = n^2$ può solo essere un limite inferiore ne stretto per $f(n) = n^2 \log n = \Omega(n^2 \log n) = \Omega(n^{\log_b a + \epsilon})$ per qualche $\epsilon > 0$. Potremmo potenzialmente essere nel caso 3 del teorema del master. Ci manca

da verificare che $af(\frac{n}{b}) = 4f(\frac{n}{2}) = 4(\frac{n}{2})^2 \log(\frac{n}{2}) \leq cf(n) = cn^2 \log n$ per qualche $c < 1$. Osserviamo che $4(\frac{n}{2})^2 \log(\frac{n}{2}) = 4\frac{n^2}{4}(\log n - 1) = n^2(\log n - 1) = n^2 \log n - n^2$ e assumiamo, per assurdo, che esista un $c < 1$ tale che

$$
\begin{aligned}
& n^2 \log n - n^2 \leq cn^2 \log n \\
\text{sse} \\
& n^2 \log n - cn^2 \log n \leq n^2 \\
\text{sse} \\
& (1-c)n^2 \log n \leq n^2 \\
\text{sse (dividendo tutto per } & (1-c)n^2 \geq 1) \\
& \log n \leq \tfrac{1}{1-c}
\end{aligned}
$$

Il che è impossibile visto che c e, quindi, anche $\frac{1}{1-c}$ è una costante.

Questo significa che la ricorrenza data non soddisfa la seconda condizione del terzo caso del teorema del master; quindi tale teorema non può essere usato per risolvere $T(n)$.