Corso di laurea in Informatica
Introduzione agli Algoritmi
Didattica blended

Esercizio risolto su equazioni di ricorrenza

Angelo Monti

SAPIENZA
UNIVERSITÀ DI ROMA

Sulla base delle slides a cura di T. Calamoneri e G. Bongiovanni per il corso di informatica generale AA 2019/2020

Esercizio: Risolvere la seguente equazione di ricorrenza con i quattro metodi visti a lezione

- $T(n) = 2 \cdot T\left(\frac{n}{2}\right) + \Theta(n)$
- $T(1) = \Theta(1)$

1) Risolvere la seguente equazione con il metodo iterativo

- $T(n) = 2 \cdot T\left(\frac{n}{2}\right) + \Theta(n)$
- $T(1) = \Theta(1)$

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

$$= 2^2 T\left(\frac{n}{2^2}\right) + 2 \cdot \Theta(n)$$

$$= 2^2\left(2T\left(\frac{n}{2^3}\right) + \Theta\left(\frac{n}{2^2}\right)\right) + 2 \cdot \Theta(n) = 2^3 T\left(\frac{n}{2^3}\right) + 3 \cdot \Theta(n)$$

$$= \ldots\ldots$$

$$= 2^k T\left(\frac{n}{2^k}\right) + k \cdot \Theta(n)$$

Esercizio:

segue metodo iterativo per $T(n) = 2T(n/2) + \Theta(n)$ e $T(1) = \Theta(1)$

$$T(n) = 2^k \cdot T\left(\frac{n}{2^k}\right) + k \cdot \Theta(n)$$

…(vado avanti finché $\frac{n}{2^k} = 1$ *cioè quando* $k = \log n$*)…*

$= 2^{\log n} T(1) + (\log n)\Theta(n)$ (ricordando che $2^{\log n} = n = \Theta(n)$)

$$= n\Theta(1) + (\log n)\Theta(n) = \Theta(n \log n)\,.$$

Dunque la soluzione tramite questo metodo è $\Theta(n \log n)$.

2) Risolvere la seguente equazione con il metodo di sostituzione

- $T(n) = 2 \cdot T\left(\frac{n}{2}\right) + \Theta(n)$
- $T(1) = \Theta(1)$

Dobbiamo innanzi tutto eliminare la notazione asintotica, quindi l'equazione diventa:

$$T(n) = 2T\left(\frac{n}{2}\right) + c \cdot n \qquad \text{per qualche costante } c > 0$$

$$T(1) = d \qquad \text{per qualche costante } d > 0$$

Proviamo a dimostrare per induzione che la soluzione è $\Omega(n \log n)$

$T(n) \geq k \cdot n \log n,$ dove k è una costante da determinare.

Esercizio:

segue metodo di sostituzione per $T(n) = 2T(n/2) + \Theta(n)$ e $T(1) = \Theta(1)$

Vogliamo dimostrare per induzione che $T(n) \geq k \cdot n \log n$, dove k è una costante da determinare.

Passo base. $T(1) \geq 0$, che è sempre verificata.

Passo induttivo. Sostituendo nell'equazione generica otteniamo:

$$
\begin{aligned}
T(n) &\geq 2\left(k \cdot \frac{n}{2} \log \frac{n}{2}\right) + c \cdot n \\
&= k \cdot n \log \frac{n}{2} + c \cdot n \\
&= k \cdot n \log n - k \cdot n + c \cdot n \\
&\geq k \cdot n \log n
\end{aligned}
$$

che è vera se $c \cdot n \geq k \cdot n$ cioè se $k \leq c$. Poiché un tale *k* è sempre possibile da trovare, ne concludiamo che

$$T(n) = \Omega(n \log n).$$

Esercizio:

segue metodo di sostituzione per $T(n) = 2T(n/2) + \Theta(n)$ e $T(1) = \Theta(1)$

Per quanto riguarda la maggiorazione, non possiamo usare $T(n) \leq k'n \log\, n$ perché in tal modo il passo base non è verificato. Tentiamo allora $T(n) \leq k'n \log\, n + h$.

Passo base. $T(1) = d \leq h$ che è vera per $h \geq d$.

Passo induttivo. Sostituendo l'ipotesi induttiva nell'equazione otteniamo:

$$
\begin{aligned}
T(n) &\leq 2\left(k'\frac{n}{2}\log\,\frac{n}{2} + h\right) + cn \\
&= k' \cdot n \log \frac{n}{2} + 2h + c \cdot n \\
&= k' \cdot n(\log\, n - 1) + 2h + c \cdot n \\
&= k' \cdot n \log\, n + h - k' \cdot n + c \cdot n + h \\
&\leq k' \cdot n \log\, n + h \qquad \textit{che è vera se } c \cdot n + h \leq k' \cdot n. \textit{ ed esempio se } k' = c + h
\end{aligned}
$$

Esercizio:

segue metodo di sostituzione per $T(n) = 2T(n/2) + \Theta(n)$ e $T(1) = \Theta(1)$

Anche in questo caso esistono dunque opportuni valori di h e k', fissati c e d, per cui la disuguaglianza è verificata.

Ne segue che: $T(n) = O(n \log n)$.

Mettendo insieme le due notazioni asintotiche trovate, e cioè $T(n) = \Omega(n \log n)$ e $T(n) = O(n \log n)$, si deduce che :

$$T(n) = \Theta(n \log n).$$

3) Risolvere la seguente equazione con il metodo principale

- $T(n) = 2T(n/2) + \Theta(n)$
- $T(1) = \Theta(1)$

Possiamo applicare questo metodo perché la ricorrenza soddisfa le ipotesi del teorema; inoltre:

- $a = 2, \ b = 2$
- $f(n) = \Theta(n)$
- $n^{\log_b a} = n^{\log_2 2} = n$

Poiché $f(n) = \Theta\left(n^{\log_b a}\right)$ siamo nel **caso 2**, da cui:

$$T(n) = \Theta(n \log n)\,.$$

4) Risolvere la seguente equazione con il metodo dell'albero:

- $T(n) = 2 \cdot T\left(\frac{n}{2}\right) + \Theta(n)$
- $T(1) = \Theta(1)$

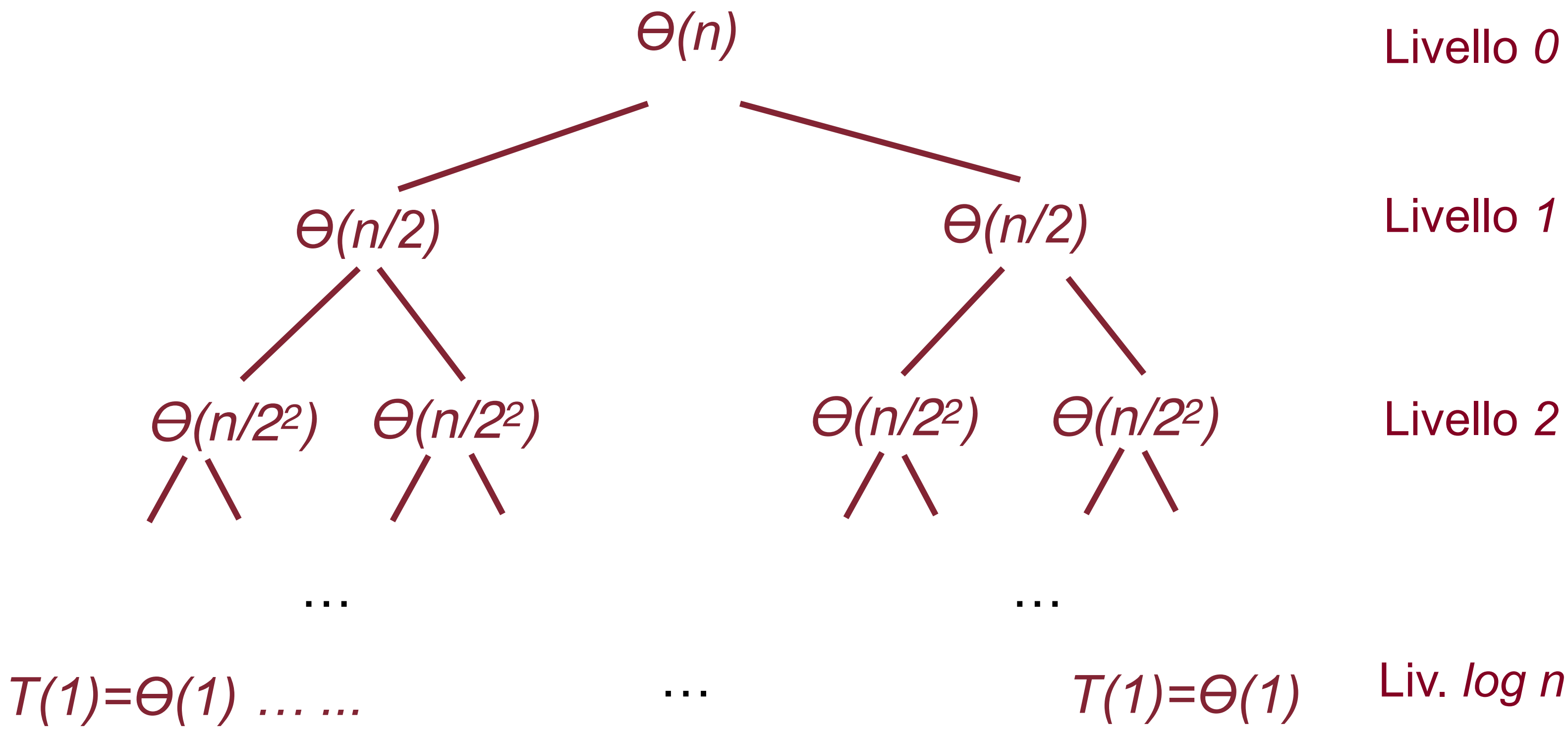
La radice dell'albero dà un contributo di $\Theta(n)$ ed ha due figli, entrambi etichettati con *T(n/2)*, che quindi danno ciascuno contributo $\Theta\left(\frac{n}{2}\right)$ e così via. Schematizziamo dunque l'albero in questo modo:

$$
\begin{array}{ccc}
 & \Theta(n) & \\
\swarrow & & \searrow \\
T(n/2) & & T(n/2)
\end{array}
$$

Esercizio:

segue metodo dell'albero per $T(n) = 2T(n/2) + \Theta(n)$ e $T(1) = \Theta(1)$

Iterando il procedimento otteniamo questo albero, nel quale numeriamo i livelli da 0 (radice) a $\log\, n$:



Esercizio:

segue metodo dell'albero per $T(n) = 2T(n/2) + \Theta(n)$ e $T(1) = \Theta(1)$

Il contributo della generica riga i-esima dell'albero è dato dal valore su ciascuno dei suoi nodi, pari a $\Theta\left(\frac{n}{2^i}\right)$, moltiplicato per il numero di nodi, cioè 2^i.

Considerato che le righe sono $\log\ n + 1$(da 0 a $\log\ n$), si ha:

$$T(n) = \sum_{i=0}^{\log\ n} \Theta\left(\frac{n}{2^i}\right) 2^i = \sum_{i=0}^{\log\ n} \Theta(n) = \Theta(n \log\ n)$$