Corso di laurea in Informatica
Introduzione agli Algoritmi

Esercizi sulle Equazioni di ricorrenza

Angelo Monti

SAPIENZA
UNIVERSITÀ DI ROMA

Esercizio 1 da risolvere con tutti e 4 i metodi

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

Metodo iterativo:

$$
\begin{aligned}
T(n) &= 2T\left(\frac{n}{2}\right) + \Theta(n) \\
&= 2\left(2T\left(\frac{n}{2}\right) + \Theta\left(\frac{n}{2}\right)\right) + \Theta(n) \\
&= 2^2T\left(\frac{n}{2^2}\right) + \Theta(n) + \Theta(n) \\
&= \ldots\ldots \\
&= 2^i\left(\frac{n}{2^i}\right) + i\Theta(n)
\end{aligned}
$$

Ci fermiamo quando $\frac{n}{2^i} = 1$ vale a dire $i = \log_2 n$ ed otteniamo ....

$$
\begin{aligned}
T(n) &= 2^{\log_2 n} T(1) + \log n \Theta\left(n\right) \\
&= n\Theta(1) + \Theta\left(n \log n\right) \\
&= \Theta(n) + \Theta\left(n \log n\right) \\
&= \Theta\left(n \log n\right)
\end{aligned}
$$

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

ESERCIZI DA FARE A CASA:

Risolvere con il metodo iterativo:

$$T(n) = 3T\left(\frac{n}{2}\right) + \Theta(n) \ \ e \ \ T(1) = \Theta(1)$$

e

$$T(n) = 2T\left(\frac{n}{3}\right) + \Theta(n) \ \ e \ \ T(1) = \Theta(1)$$

Utili per notare come cambiano i conti al variare delle costanti moltiplicative

Esercizio 1

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

Metodo di sostituzione:

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

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

$$T(1) = a$$

Dove a e b sono costanti positive.

Proviamo a dimostrare per induzione che la soluzione è:

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

Passo base. $T(1) \geq a \geq c1\log_2 1 = 0$, che è sempre verificata.

Passo induttivo. otteniamo:

$$\begin{aligned} T(n) &\geq 2c\frac{n}{2}\log_2\frac{n}{2} + b \cdot n \\ &= c \cdot n(\log_2 n - 1) + b \cdot n \\ &= c \cdot n\log_2 n - c \cdot n + b \cdot n \\ &\geq c \cdot n\log_2 n \end{aligned}$$

Dove l'ultima diseguaglianza segue se $c \leq b$.

Abbiamo dimostrato che $T(n) = \Omega(n\log n)$.

Per quanto riguarda la maggiorazione, non possiamo usare $T(n) \geq cn \log n$ perché in tal modo il passo base non è verificato.

Tentiamo allora $T(n) \leq cn \log_2 n + d$.
<u>*Passo base*</u>. $T(1) = a \leq c * 1 * 0 + d$ che è vera per $d \geq a$.
<u>*Passo induttivo*</u>.

$$\begin{aligned} T(n) &\leq 2\left(c\frac{n}{2}\log_2\frac{n}{2} + d\right) + b \cdot n \\ &= c \cdot n(\log_2 n - 1) + 2d + b \cdot n \\ &= c \cdot n \log_2 n + d - c \cdot n + b \cdot n + d \\ &\leq c \cdot n \log_2 n + d \end{aligned}$$

Dove l'ultima diseguaglianza segue se $b \cdot n + d \leq c \cdot n$ che è verificata ad esempio per $c = b + d$.
Abbiamo dimostato 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 $T(n) = \Theta(n \log n)$.

Esercizio 1

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

Metodo di principale:

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

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

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

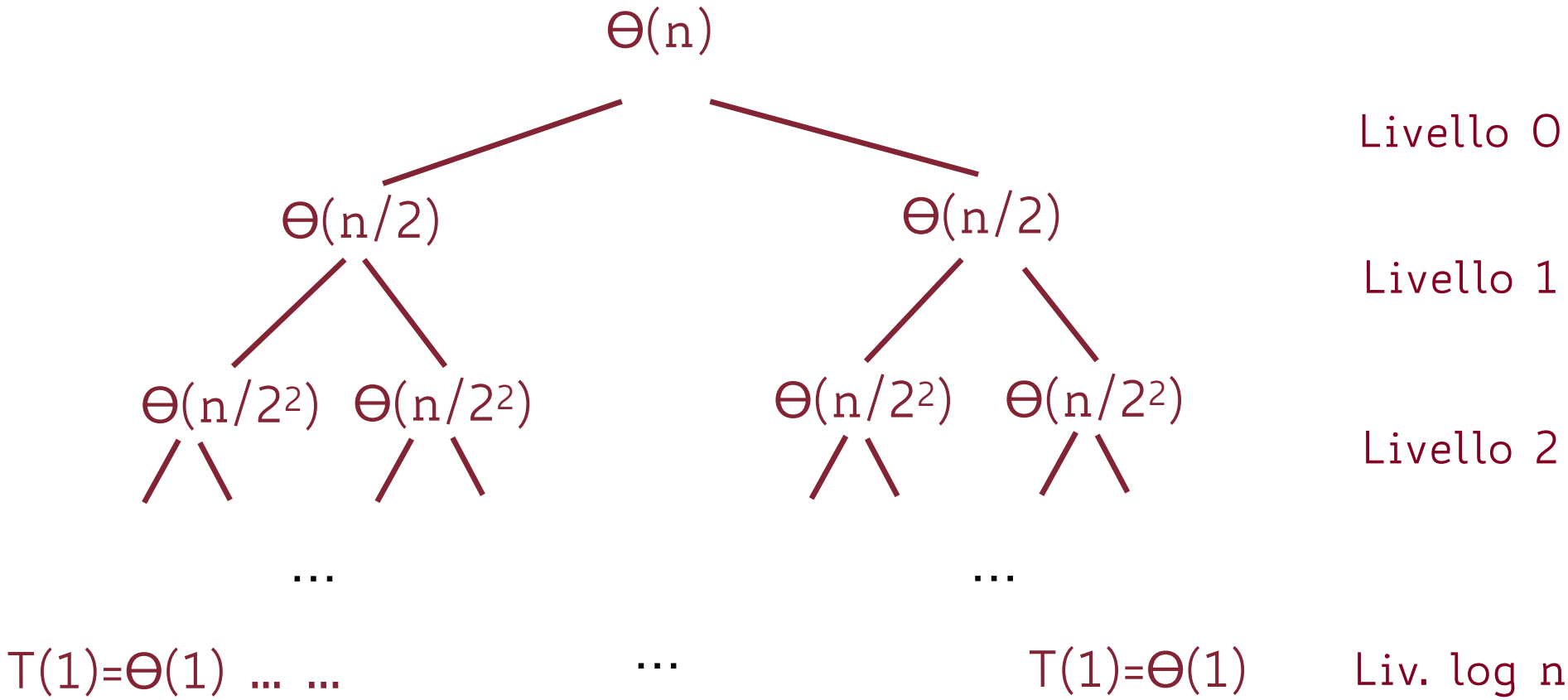
$$T(n) = \Theta(n \log n)$$

Esercizio 1

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

Metodo dell'albero:

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



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 della riga, cioè 2^i. Considerato che le righe sono $\log n + 1$ (da 0 a $\log n$) si ha:

$$T(n) = \sum_{i=0}^{\log n} \Theta(n) = \Theta(n \log n)$$

Esercizio 2

- $T(n) = 4T\left(\frac{n}{2}\right) + \Theta(n)$
- $T(1) = \Theta(1)$

Metodo iterativo:

$$
\begin{aligned}
T(n) &= 4T\left(\frac{n}{2}\right) + \Theta(n) \\
&= 4\left(4T\left(\frac{n}{2^2}\right) + \Theta\left(\frac{n}{2}\right)\right) + \Theta(n) \\
&= 4^2T\left(\frac{n}{2^2}\right) + 4\Theta\left(\frac{n}{2}\right) + \Theta(n) \\
&= \ldots\ldots \\
&= 4^iT\left(\frac{n}{2^i}\right) + \sum_{j=0}^{i-1} 4^j\Theta\left(\frac{n}{2^j}\right)
\end{aligned}
$$

Ci fermiamo quando $\frac{n}{2^i} = 1$ vale a dire $i = \log_2 n$ ed otteniamo ....

$$
\begin{aligned}
T(n) &= 4^{\log_2 n} T(1) + \Theta\left(n \sum_{j=0}^{\log n - 1} 2^j\right) \\
&= 2^{2\log_2 n}\Theta(1) + \Theta\left(n 2^{\log n}\right) \\
&= \Theta(n^2) + \Theta\left(n^2\right) \\
&= \Theta\left(n^2\right)
\end{aligned}
$$

Dunque la soluzione tramite questo metodo è $\Theta(n^2)$.

Esercizio 2

- $T(n) = 4T\left(\frac{n}{2}\right) + \Theta(n)$
- $T(1) = \Theta(1)$

Metodo principale:

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

- $a = 4, \quad b = 2$
- $n^{\log_b a} = n^{\log_2 4} = n^2$
- $f(n) = \Theta(n) = O\left(n^{\log_b a - \epsilon}\right)$ (ad es. per $\epsilon = 1$)

Siamo quindi nel **caso 1**, da cui: $T(n) = \Theta(n^2)$.

Esercizio 2

- $T(n) = 4T\left(\frac{n}{2}\right) + \Theta(n)$
- $T(1) = \Theta(1)$

Metodo di sostituzione:

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

$$T(n) = 4T\left(\frac{n}{2}\right) + b \cdot n$$

$$T(1) = a$$

Dove a e b sono costanti positive. Proviamo a dimostrare per induzione che la soluzione è:

$T(n) \geq c \cdot n^2$, dove c è una costante da determinare.

Passo base. $T(1) = a \geq c$, da cui deduciamo una prima condizione su c.

Passo induttivo. Sostituendo nell'equazione generica otteniamo:

$$\begin{aligned} T(n) &\geq 4c\left(\frac{n}{2}\right)^2 + b \cdot n \\ &= c \cdot n^2 + b \cdot n \\ &\geq c \cdot n^2 \end{aligned}$$

Ne concludiamo che $T(n) = \Omega(n^2)$.

Per quanto riguarda la maggiorazione, tentiamo la soluzione $T(n) \leq c \cdot n^2$ dove c è una costante da determinare.

<u>*Passo base*</u>. $T(1) = a \leq c$, da cui deduciamo una prima condizione su c.

<u>*Passo induttivo*</u>. Sostituendo nell'equazione generica otteniamo:

$T(n) \leq 4c\left(\frac{n}{2}\right)^2 + b \cdot n = c \cdot n^2 + b \cdot n$ che non è mai $\leq c \cdot n^2$ perché b è una costante positiva.

Tentiamo allora $T(n) \leq c \cdot n^2 - d \cdot n$.

<u>*Passo base*</u>. $T(1) = d \leq c - d$ che è vera per certi valori di *c* e *d*.

<u>*Passo induttivo*</u>. Sostituendo l'ipotesi induttiva nell'equazione otteniamo:

$$\begin{aligned} T(n) &\leq 4\left(c\left(\frac{n}{2}\right)^2 - d\left(\frac{n}{2}\right)\right) + b \cdot n \\ &= c \cdot n^2 - 2d \cdot n + b \cdot n \\ &= c \cdot n^2 - d \cdot n - d \cdot n + b \cdot n \\ &\leq c \cdot n^2 - d \cdot n \end{aligned}$$

Dove l'ultima diseguaglianza segue prendendo $d \geq b$.

Ne concludiamo che $T(n) = O(n^2)$ e, unendo i due risultati abbiamo: $T(n) = \Theta(n^2)$.

Esercizio 2

- $T(n) = 4T\left(\frac{n}{2}\right) + \Theta(n)$
- $T(1) = \Theta(1)$

Metodo dell'albero:

Nell'albero ogni nodo ha 4 figli questo significa che a livello i ci saranno 4^i nodi ciascuno dei quali contribuirà per un costo $\Theta\left(\frac{n}{2^i}\right)$.

La ricorsione termina quando $\frac{n}{2^i} = 1$, a dire $i = \log_2 n$ questo significa che i livelli dell'albero sono $\log n + 1$ (da 0 a $\log n$).

Il contributo totale sommato per livelli sarà

$$\sum_{i=0}^{\log n} 4^i \Theta\left(\frac{n}{2^i}\right) = \sum_{i=0}^{\log n} 2^i \Theta n) = \Theta(n) \sum_{i=0}^{\log n} 2^i = \Theta(n)\Theta\left(2^{\log n}\right) = \Theta(n)\Theta(n) = \Theta(n^2)$$

Dove per risolvere la sommatoria ho utilizzato

$$\sum_{i=0}^{x} 2^i = \Theta\left(2^x\right)$$

Esercizio 3

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

Metodo iterativo:

$$
\begin{aligned}
T(n) &= 2T\left(\frac{n}{2}\right) + \Theta(n^2) \\
&= 2\left(2T\left(\frac{n}{2^2}\right) + \Theta\left(\left(\frac{n}{2}\right)^2\right)\right) + \Theta(n^2) \\
&= 2^2T\left(\frac{n}{2^2}\right) + 2\Theta\left(\left(\frac{n}{2}\right)^2\right) + \Theta(n^2) \\
&= \ldots\ldots \\
&= 2^iT\left(\frac{n}{2^i}\right) + \sum_{j=0}^{i-1} 2^j\Theta\left(\left(\frac{n}{2^j}\right)^2\right) \\
&= 2^iT\left(\frac{n}{2^i}\right) + \Theta\left(n^2\sum_{j=0}^{i-1}\frac{1}{2^j}\right)
\end{aligned}
$$

Ci fermiamo quando $\frac{n}{2^i} = 1$ vale a dire $i = \log_2 n$ ed otteniamo ....

$$
\begin{aligned}
T(n) &= 2^{\log_2 n} T(1) + \Theta\left(n^2 \sum_{j=0}^{\log n - 1} \frac{1}{2^j}\right) \\
&= n\Theta(1) + \Theta(n^2)\Theta(1) \\
&= \Theta(n) + \Theta(n^2) \\
&= \Theta\left(n^2\right)
\end{aligned}
$$

Dove ho utilizzato $\sum_{i=0}^{x} c^i = \Theta(1)$ *quando* $c < 1$

Dunque la soluzione tramite questo metodo è $\Theta(n^2)$.

Esercizio 3

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

Metodo principale:

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

- $a = 2,\ b = 2$
- $n^{\log_b a} = n^{\log_2 2} = n$
- $f(n) = \Theta(n^2) = \Omega(n^{\log_b a + \epsilon})$ (ad es. per ε=1)

Poiché $af\left(\frac{n}{b}\right) = 2\left(\frac{n}{2}\right)^2 = \frac{n^2}{2} < cn^2$ con $c = \frac{1}{2}$, siamo nel caso 3 e possiamo concludere che $T(n) = \Theta(n)$

Esercizio 3

- $T(n) = 4T\left(\frac{n}{2}\right) + \Theta(n^2)$
- $T(1) = \Theta(1)$

Metodo di sostituzione:

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

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

$$T(1) = a$$

Per a e b costanti positive.

Proviamo a dimostrare per induzione la soluzione:

$T(n) \leq cn^2$ dove c è una costante da determinare.

Passo base. $T(1) \leq a \leq c$, da cui deduciamo una prima condizione su c.

Passo induttivo. Sostituendo nell'equazione generica otteniamo:

$$\begin{aligned} T(n) &\leq 2c\left(\frac{n}{2}\right)^2 + b \cdot n^2 \\ &= c\frac{n^2}{2} + b \cdot n^2 \\ &= \left(\frac{c}{2} + b\right) n^2 \\ &\leq c \cdot n^2 \end{aligned}$$

Dove l'ultima disuguaglianza vale se $\frac{c}{2} + b \leq c$ e basta quindi prendere $c \geq 2b$.

Abbiamo dimostrato che $T(n) = O(n^2)$

Proviamo ora a dimostrare per induzione che vale anche $T(n) = \Omega(n^2)$. Ipotizziamo dunque:

$T(n) \geq c \cdot n^2$ dove c è una costante da determinare.

<u>*Passo base*</u>. $T(1) = a \geq c$, da cui deduciamo una prima condizione su c.

<u>*Passo induttivo*</u>. Sostituendo nell'equazione generica otteniamo:

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

Dove l'ultima disuguaglianza vale se $\frac{c}{2} + b \geq c$ e basta quindi prendere $c \leq 2b$.

Ne concludiamo che $T(n) = \Omega(n^2)$

Dalle due deduciamo $T(n) = \Theta(n^2)$

Esercizio 3

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

Metodo dell'albero:

Nell'albero ogni nodo ha 2 figli questo significa che a livello i ci saranno 2^i nodi ciascuno dei quali contribuirà per un costo $\Theta\left(\left(\frac{n}{2^i}\right)^2\right)$. La ricorsione termina quando $\frac{n}{2^i} = 1$, a dire $i = \log_2 n$ questo significa che i livelli dell'albero sono $\log n + 1$ (da 0 a $\log n$). Il contributo totale sommato per livelli sarà:

Il contributo totale sommato per livelli sarà

$$\sum_{i=0}^{\log n} 2^i \Theta\left(\left(\frac{n}{2^i}\right)^2\right) = \sum_{i=0}^{\log n} \Theta\left(\frac{n^2}{2^i}\right) = \Theta\left(n^2 \sum_{i=0}^{\log n} \frac{1}{2^i}\right) = \Theta(n^2)$$

Dove per risolvere la sommatoria ho utilizzato $\sum_{i=0}^{x} c^i = \Theta(1)$ *quando* $c < 1$

Esercizio 4

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

Metodo principale:

$a = b = 2$ e $n^{\log_b a} = n$

$n^{\log_b a}$ = n è asintoticamente più piccolo di $f(n)$, ma non polinomialmente. Di conseguenza **non** possiamo applicare il metodo del teorema principale.

Esercizio 4

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

Metodo iterativo:

$$
\begin{aligned}
T(n) &= 2T\left(\frac{n}{2}\right) + \Theta(n \log n) \\
&= 2\left(2T\left(\frac{n}{2}\right) + \Theta\left(\frac{n}{2}\log\frac{n}{2}\right)\right) + \Theta(n \log n) \\
&= 2^2 T\left(\frac{n}{2^2}\right) + \Theta\left(n \log\frac{n}{2}\right) + \Theta(n \log n) \\
&= \ldots\ldots \\
&= 2^i\left(\frac{n}{2^i}\right) + \Theta\left(\sum_{j=0}^{i-1} n \log\frac{n}{2^j}\right) \\
&= 2^i\left(\frac{n}{2^i}\right) + \Theta\left(n\sum_{j=0}^{i-1}(\log n - \log 2^j)\right) \\
&= 2^i\left(\frac{n}{2^i}\right) + \Theta\left(n\sum_{j=0}^{i-1}(\log n - j)\right)
\end{aligned}
$$

Ci fermiamo quando $\frac{n}{2^i} = 1$ vale a dire $i = \log_2 n$ ed otteniamo ....

$$
\begin{aligned}
T(n) &= 2^{\log_2 n} T(1) + \Theta\left(n \sum_{j=0}^{\log n - 1} (\log n - j)\right) \\
&= n\Theta(1) + \Theta\left(n \sum_{k=1}^{\log n} k\right) \\
&= \Theta(n) + \Theta\left(n \log^2 n\right) \\
&= \Theta\left(n \log^2 n\right)
\end{aligned}
$$

Dove ho usato che $\sum_{k=1}^{x} k = \Theta\left(x^2\right)$

Esercizio 4

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

Metodo di sostituzione:

Impostiamo la dimensione del caso base a 2, per evitare di dover gestire il caso di log 1 = 0 ➡ T(2)= Θ(1).

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

- $T(n) = 2T\left(\frac{n}{2}\right) + bn \log n$
- $T(2) = a$

dove a e b sono costanti positive

Proviamo a dimostrare per induzione la soluzione:

$T(n) \leq cn \log_2 n$, dove c è una costante da determinare.

Passo base. $T(2) = a \leq c * 2 * 1$, che è vera per $c \geq a/2$.

Passo induttivo:

$$
\begin{aligned}
T(n) &= 2T\left(\frac{n}{2}\right) + b \cdot n \log_2 n \\
&\leq 2c \cdot \frac{n}{2} \log_2^2 \frac{n}{2} + b \cdot n \log_2 n \\
&\leq c \cdot n(\log_2 - 1)^2 + b \cdot n \log_2 n \\
&\leq c \cdot n(\log_2 + 1 - 2\log_2 n) + b \cdot n \log_2 n \\
&\leq c \cdot n \log_2 + c \cdot n - 2c \cdot n \log_2 n + b \cdot n \log_2 n \\
&\leq c \cdot n \log_2 - c \cdot n \log_2 n + b \cdot n \log_2 n \\
&\leq c \cdot n \log_2
\end{aligned}
$$

Dove la penultima diseguaglianza vale perché $cn \leq cn \log n$ e per l'ultima diseguaglianza basta prendere $c \geq b$

Abbiamo quindi dimostrato che $T(n) = O(n \log^2 n)$

Si lascia per esercizio dimostrare che $T(n) = \Omega(n \log^2 n)$

Esercizio 5

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

Metodo principale:

$a = b = 2$ e $n^{\log_b a} = n$

$n^{\log_b a}$ = n è asintoticamente più grande di $f(n)$, ma non polinomialmente. Di conseguenza **non** possiamo applicare il metodo del teorema principale.

Esercizio 5

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

Metodo iterativo:

$$
\begin{aligned}
T(n) &= 2T\left(\frac{n}{2}\right) + \Theta\left(\frac{n}{\log n}\right) \\
&= 2\left(2T\left(\frac{n}{2}\right) + \Theta\left(\frac{n/2}{\log n/2}\right)\right) + \Theta\left(\frac{n}{\log n}\right) \\
&= 2^2 T\left(\frac{n}{2^2}\right) + \Theta\left(\frac{n}{\log_2 n/2}\right) + \Theta\left(\frac{n}{\log n}\right) \\
&= \ldots\ldots \\
&= 2^i\left(\frac{n}{2^i}\right) + \Theta\left(\sum_{j=0}^{i-1} \frac{n}{\log \frac{n}{2^j}}\right) \\
&= 2^i\left(\frac{n}{2^i}\right) + \Theta\left(n\sum_{j=0}^{i-1} \frac{1}{\log n - j}\right)
\end{aligned}
$$

Ci fermiamo quando $\frac{n}{2^i} = 1$ vale a dire $i = \log_2 n$ ed otteniamo ....

$$
\begin{aligned}
T(n) &= 2^{\log_2 n} T(1) + \Theta\left(n \sum_{j=0}^{\log n - 1} \frac{1}{\log n - j}\right) \\
&= n\Theta(1) + \Theta\left(n \sum_{k=1}^{\log n} \frac{1}{k}\right) \\
&= \Theta(n) + \Theta\left(n \log\log n\right) \\
&= \Theta\left(n \log\log n\right)
\end{aligned}
$$

Dove ho usato che $\sum_{k=1}^{x} \frac{1}{k} = \Theta(\log x)$

Corso di laurea in Informatica
Introduzione agli Algoritmi
A.A. 2023/24

Esercizi per casa

SAPIENZA
UNIVERSITÀ DI ROMA

Risolvere con il metodo iterativo: e con il teorema principale

$$T(n) = 3T\left(\frac{n}{2}\right) + \Theta(n) \ \ e \ \ T(1) = \Theta(1)$$

e

$$T(n) = 2T\left(\frac{n}{3}\right) + \Theta(n) \ \ e \ \ T(1) = \Theta(1)$$

Utili per notare come cambiano i conti al variare delle costanti moltiplicative

- Risolvere l'equazione:

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n \log n);$$

$$T(1) = \Theta(1)$$

con il metodo dell'albero.

- Risolvere l'equazione

$$T(n) = 2T\left(\frac{n}{2}\right) + \Theta\left(\frac{n}{\log n}\right);$$

- $T(1) = \Theta(1)$
- con il metodo dell'albero.

- Calcolare l' equazione di ricorrenza associata al seguente algoritmo e risolverla con tutti i metodi possibili:

```
def Test (n)
  k = 0
  for i in range(1, n):
     k = k + 1
  if n ≤ 1: return k
      else: return (Test(n DIV 2)+Test(n DIV 4))
```

Esercizi (2)

- Calcolare l' equazione di ricorrenza associata al seguente algoritmo e risolverla con tutti i metodi possibili:

```
def Test (n)
  k = 0
  for i in range(1, n):
     k = k + 1
  if n ≤ 1: return k
      else: return (Test(n DIV 2)+Test(n DIV 4))
```