# Corso di laurea in Informatica
# Introduzione agli Algoritmi
# Didattica blended

## Esercizi sulle 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 con il metodo iterativo

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

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

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

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

$$= \ldots\ldots$$

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

### 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* $k = \log n$*)*…

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

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

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

ESERCIZI DA FARE A CASA:

Risolvere con il metodo iterativo:

$T(n) = 3T(n/2) + \Theta(n)$ *e* $T(1) = \Theta(1)$

e

$T(n) = 2T(n/3) + \Theta(n)$ *e* $T(1) = \Theta(1)$

Notate come cambiano i conti al variare delle costanti moltiplicative

## Esercizio: Risolvere la seguente equazione con il metodo di sostituzione

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

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

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

$$T(1) = a \quad \text{per qualche costante } a > 0$$

Proviamo a dimostrare per induzione che la soluzione è:
$T(n) \geq c \cdot n \log n + d,$ dove $c$ e $d$ sono costanti 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 $T(n) \geq c \cdot n \log n + d,$ dove $c$ e $d$ sono costanti da determinare.

*Passo base.* $T(1) \geq a \geq d$, che è verificata prendendo $d \leq a$.
*Passo induttivo.* Sostituendo nell'equazione generica otteniamo:

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

che è vera se $-c \cdot n + b \cdot n + d \geq 0$ cioè se $c \leq \frac{bn + d}{n}$. Prendendo dunque ad esempio $d = a$ e $c = b$, 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, assumiamo $T(n) \leq cn \log n + d.$

*Passo base.* $T(1) = b \leq d$ che è vera per $d \geq d.$
*Passo induttivo.* Sostituendo l'ipotesi induttiva nell'equazione otteniamo:

$$\begin{aligned} T(n) &\leq 2\left(c\frac{n}{2} \log \frac{n}{2} + d\right) + bn \\ &= cn \log \frac{n}{2} + 2d + bn \\ &= cn \log n - cn + 2d + b \cdot n \\ &= cn \log n + d - cn + bn + d \\ &\leq cn \log n + d \quad \textit{che è vera se } -cn + bn + d \leq 0. \textit{ ed esempio se } c \geq \frac{bn + d}{n} \end{aligned}$$

**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 $c$ e $d$, fissati $a$ e $b$, 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 :

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

**Esercizio: 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:

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

ESERCIZI DA FARE A CASA:

Risolvere con il metodo principale:

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

*e*

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

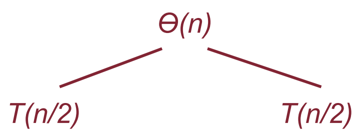
Notate come cambia il caso del teorema da applicare al variare di $a$ e $b$

**Esercizio: 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.

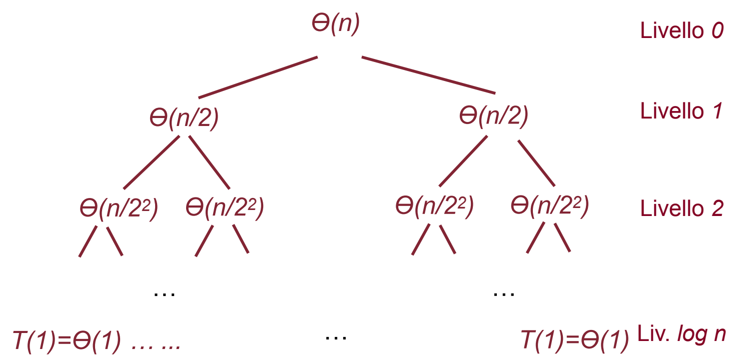
Schematizziamo dunque l'albero in questo modo:




## 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:

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

- **Esercizio: risolvere la seguente equazione con il metodo iterativo:**
- $T(n) = 4 \cdot T\left(\frac{n}{2}\right) + \Theta(n)$
- $T(1) = \Theta(1)$

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

$$= 4^2T\left(\frac{n}{2^2}\right) + 4^1\Theta\left(\frac{n}{2^1}\right) + \Theta(n) = 4^2\left(4T\left(\frac{n}{2^3}\right) + \Theta\left(\frac{n}{2^2}\right)\right) + 4^1\Theta\left(\frac{n}{2^1}\right) + \Theta(n^1)$$

$$= 4^3T\left(\frac{n}{2^3}\right) + 4^2\Theta\left(\frac{n}{2^2}\right) + 4^1\Theta\left(\frac{n}{2^1}\right) + \Theta(n^1)$$

$$= \ldots\ldots$$

$$= 4^kT\left(\frac{n}{2^k}\right) + \sum_{i=0}^{k-1} 4^i\Theta\left(\frac{n}{2^i}\right)$$

**ESERCIZIO:**

segue metodo iterativo per $T(n) = 4T(n/2) + \Theta(n)$ e $T(1) = \Theta(1)$

$$T(n) = 4^kT\left(\frac{n}{2^k}\right) + \sum_{i=0}^{k-1} 4^i\Theta\left(\frac{n}{2^i}\right) = 4^kT\left(\frac{n}{2^k}\right) + \sum_{i=0}^{k-1} 2^i\Theta(n)$$

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

$$= 4^{\log n}T(1) + \Theta(n) \sum_{i=0}^{\log n-1} 2^i \quad \textit{(ora: } 4^{\log n} = n^2 \textit{ e } \sum_{i=0}^{\log n-1} 2^i = \Theta\left(2^{\log n}\right) = \Theta(n))$$

$$= n^2\Theta(1) + \Theta(n)\Theta(n)$$

$$= \Theta(n^2).$$

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

- **Esercizio: risolvere la seguente equazione con il metodo principale:**
- $T(n) = 4T\left(\frac{n}{2}\right) + \Theta(n)$
- $T(1) = \Theta(1)$

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

- $a = 4, \ b = 2$
- $n^{\log_b a} = n^{\log_2 4} = n^2$
- $f(n) = \Theta(n) = O\left(n^{\log_b a - \epsilon}\right)$ (ad es. per $\epsilon = 0.5$)

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

- **Esercizio: risolvere la seguente equazione con il metodo di sostituzione**
- $T(n) = 4T\left(\frac{n}{2}\right) + \Theta(n)$
- $T(1) = \Theta(1)$

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

$T(n) = 4T\left(\frac{n}{2}\right) + b \cdot n$ per qualche costante b>0

$T(1) = a$ per qualche costante a>0

Proviamo a dimostrare per induzione che la soluzione è:

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

**ESERCIZIO:**

segue metodo di sostituzione per $T(n) = 4T(n/2) + \Theta(n)$ e $T(1) = \Theta(1)$

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

*Passo induttivo.* Sostituendo nell'equazione generica otteniamo:

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

Quindi ad esempio prendendo $c = a$ possiamo concludere che $T(n) = \Omega(n^2)$.

**ESERCIZIO:**

segue metodo di sostituzione per $T(n) = 4T(n/2) + \Theta(n)$ e $T(1) = \Theta(1)$

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

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

*Passo induttivo.* Sostituendo nell'equazione generica otteniamo:

$T(n) \leq 4c\left(\frac{n}{2}\right)^2 + b \cdot n = c \cdot n^2 + b \cdot n \geq c \cdot n^2$ dove l'ultima diseguaglianza vale sempre perché *c* è una costante positiva.

**ESERCIZIO:**

segue metodo di sostituzione per $T(n) = 4T(n/2) + \Theta(n)$ e $T(1) = \Theta(1)$

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

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

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

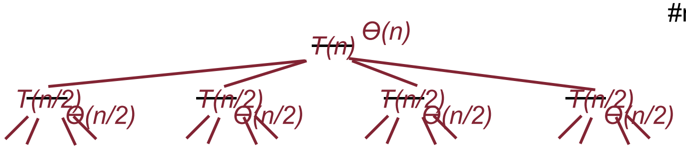
$$
\begin{aligned}
T(n) &\leq 4\left(c\left(\frac{n}{2}\right)^2 - d\frac{n}{2}\right) + b \cdot n \\
&= c \cdot n^2 - 2 \cdot d \cdot n + b \cdot n \\
&= (cn^2 - d \cdot n) - d \cdot n + b \cdot n \\
&\leq c \cdot n^2 - d \cdot n
\end{aligned}
$$

l'ultima diseguaglianza vale se $-d \cdot n + b \cdot n \leq 0$, che è verificata per $d \geq b$.

*Quindi prendendo d=b e c=a+b*

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

- **Risolvere la seguente equazione col metodo dell'albero:**
- $T(n) = 4T(n/2) + \Theta(n)$
- $T(1) = \Theta(1)$




…

| #nodi | contributo |
| --- | --- |
| 1 | $\Theta(n)$ |
| 4 | $\Theta(n/2)$ |
| … | … |
| $4^i$ | $\Theta(n/2^i)$ |
| … | … |

finché $\dfrac{n}{2^i} = 1$ cioé $i = \log n$

Sommo tutti i contributi:

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

- **Risolvere la seguente equazione col metodo iterativo:**
- $T(n) = 2T(n/2) + \Theta(n^2)$
- $T(1) = \Theta(1)$

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

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

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

……

$$= 2^kT\left(\frac{n}{2^k}\right) + \sum_{i=0}^{k-1} \frac{1}{2^i}\Theta(n^2)$$

**ESERCIZIO:**

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

$$T(n) = 2^kT\left(\frac{n}{2^k}\right) + \sum_{i=0}^{k-1} \frac{1}{2^i}\Theta(n^2)$$

……. Ci fermiamo quando $\dfrac{n}{2^k} = 1$ *cioè* $k = \log n$, *ottenendo*

$= 2^{\log n}T(1) + \Theta(n^2) \sum_{i=0}^{\log n-1} \left(\frac{1}{2}\right)^i$ *Ricordando che* $2^{\log n} = n$ *e che* $\sum_{i=0}^{\log n-1} \left(\frac{1}{2}\right)^i = \Theta(1)$

$= \Theta(n) + \Theta(n^2)$

$= \Theta(n^2)$

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

- **Risolvere la seguente equazione col metodo principale:**
- $T(n) = 2T\left(\dfrac{n}{2}\right) + \Theta(n^2)$
- $T(1) = \Theta(1)$

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

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

Siamo quindi nel **caso 3.** Poiché $a\left(\dfrac{n}{b}\right) = 2\left(\dfrac{n}{2}\right) \leq c \cdot n^2 = \dfrac{1}{2}n^2$(ad es. per $c = \dfrac{1}{2}$)

si ha:

$$T(n) = \Theta(n^2).$$

- **Risolvere la seguente equazione col metodo di sostituzione:**
- $T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n^2)$
- $T(1) = \Theta(1)$

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

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

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

Proviamo a dimostrare per induzione che la soluzione è:

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

**ESERCIZIO:**

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

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

*Passo induttivo.* Sostituendo nell'equazione generica otteniamo:

$$T(n) \leq 2 \cdot c \cdot \left(\frac{n}{2}\right)^2 + b \cdot n^2$$
$$= c \cdot \frac{n^2}{2} + b \cdot n^2$$
$$= \left(\frac{c}{2} + b\right) n^2$$
$$\leq c \cdot n^2 \text{ se vale } c \geq 2b$$

Prendendo dunque $c = max(a, 2b)$ concludiamo che $T(n) = O(n^2)$

**ESERCIZIO:**

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

Proviamo ora 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 $c$.

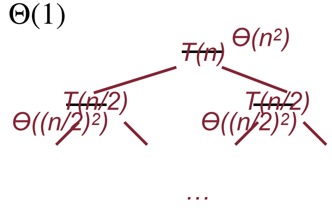
*Passo induttivo.* Sostituendo nell'equazione generica otteniamo:

$$T(n) \geq 2c\left(\frac{n}{2}\right)^2 + b \cdot n^2$$
$$= c\frac{n^2}{2} + b \cdot n^2$$
$$= \left(\frac{c}{2} + b\right) n^2$$
$$\geq c \cdot n^2 \qquad \textit{che vale se } \frac{c}{2} + b \geq c, \textit{ ossia } c \leq 2b.$$

Prendendo dunque $c = \min(a, 2b)$ concludiamo che $T(n) = \Omega(n^2)$

Da $T(n) = O(n^2)$ e $T(n) = \Omega(n)$ deduciamo $T(n) = \Theta(n^2)$

- risolvere la seguente equazione col metodo dell'albero:
- $T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n^2)$
- $T(1) = \Theta(1)$




| #nodi | contributo |
|---|---|
| $2^0$ | $\Theta(n^2)$ |
| $2^1$ | $\Theta\left(\left(\frac{n}{2}\right)^2\right)$ |
| … | … |
| $2^i$. | $\Theta\left(\left(\frac{n}{2^i}\right)^2\right)$ |
| … | … |

finché $n/2^i = 1$ cioè $i = \log n$

Sommo tutti i contributi:

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

- **Risolvere la seguente equazione di ricorrenza col metodo iterativo**
- $T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n \log n)$
- $T(1) = \Theta(1)$

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

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

$$= 2\left[2\left(2T\left(\frac{n}{2^3}\right) + \Theta\left(\frac{n}{4}\log\frac{n}{4}\right)\right) + \Theta\left(\frac{n}{2}\log\frac{n}{2}\right)\right] + \Theta(n \log n)$$

............

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

**ESERCIZIO:**

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

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

…….. Ci fermiamo quando $k = \log n$ e otteniamo:

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

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

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

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

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

$$= \Theta(n \log^2 n)$$

- **Risolvere la seguente equazione di ricorrenza col metodo di sostituzione:**
- $T(n) = 2T\left(\frac{n}{2}\right) + \Theta(n \log n)$
- $T(1) = \Theta(1)$

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

$T(n) = 2T\left(\frac{n}{2}\right) + b \cdot n \log n$ per qualche costante a > 0

$T(1) = a$ per qualche costante b > 0

**ESERCIZIO:**

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

Proviamo a dimostrare per induzione che la soluzione è:

$T(n) \leq c \cdot n \log^2 n$ +d, dove *c* e *d* sono costanti da determinare.

*Passo base.* $T(1) = a \leq d$, vera per $d \geq a$.

*Passo induttivo.* Sostituendo nell'equazione generica otteniamo:

$$T(n) = 2c\frac{n}{2}\log^2\frac{n}{2} + bn \log n$$

$$= c(\log n - 1)^2 + bn \log n$$

$$= cn(\log^2 n - 2\log n + 1) + bn \log n$$

$$= cn \log^2 n - 2cn \log n + cn + bn \log n$$

$\leq cn \log^2 n - cn \log n + bn \log n$ (perché $cn - cn \log n \leq 0$)

$\leq c \cdot n \log^2 n$ che è vera per $-c + b \leq 0$ vale a dire $c \geq b$

Quindi prendendo $d \geq a$ e $c \geq b$ Concludiamo che $T(n) = O(n \log^2)$

***Si lascia come esercizio** provare che* $T(n) = \Omega(n \log^2 n)$

- risolvere la seguente equazione col metodo principale se possibile:
- $T(n) = 2T\left(\frac{n}{2}\right) + \Theta\left(\frac{n}{\log n}\right)$
- $T(1) = \Theta(1)$

Nota che $f(n) = \Theta\left(\frac{n}{\log n}\right) = O\left(n^{\log_b a}\right) = O(n)$

Apparentemente quindi siamo nel caso 1 del metodo tuttavia $n^{\log_b a}$ pur dominando $f(n)$ non la domina polinomialmente e quindi non possiamo applicare il metodo.

- risolvere la seguente equazione col metodo iterativo:
- $T(n) = 2T\left(\frac{n}{2}\right) + \Theta\left(\frac{n}{\log n}\right)$
- $T(1) = \Theta(1)$

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

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

Ci fermiamo quando $k = \log n$ e otteniamo

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

$$= n\cdot\Theta(1) + \Theta(n)\sum_{j=1}^{\log n}\frac{1}{j}$$

Dove ho effettuato il cambio di indice $j = \log n - i$

$$= n\cdot\Theta(1) + \Theta(n)\cdot\Theta(\log\log n)$$

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

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

**Risolvere per esercizio l'equazione di ricorrenza con il metodo dell'albero ed il metodo di sostituzione**

Corso di laurea in Informatica
Introduzione agli Algoritmi
Didattica blended

Esercizi per casa

SAPIENZA
UNIVERSITÀ DI ROMA

## Esercizio:

- Calcolare l' equazione di ricorrenza associata al seguente algoritmo ricorsivo e risolverla con tutti i metodi possibili.
- L'algoritmo è invocato con $a = 0 \quad e \quad b = n - 1$ dove $n$ è la lunghezza della lista:

```
def Palindromo(lista, a,b)
    if b ≤ a :
        return True
    if lista[a]!= lista[b]:
        return False
    return Palindromo(lista, a+1, b-1)
```

## Esercizio:

- 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+1):
        k += 1
    if n ≤ 1:
        return k
    return Test(n//2)+ Test(n //4)
```