UNITELMA SAPIENZA

SAPIENZA
Università di Roma
Dipartimento di Informatica

Corso di Introduzione agli algoritmi

Prof.ssa Tiziana Calamoneri

Equazioni di ricorrenza

Equazioni di ricorrenza (1)

- Valutare il costo computazionale di un algoritmo ricorsivo è, in genere, più laborioso che nel caso degli algoritmi iterativi.
- Infatti, la natura ricorsiva della soluzione algoritmica dà luogo a una funzione di costo che, essendo strettamente legata alla struttura dell'algoritmo, è anch'essa ricorsiva.
- Trovare la funzione di costo ricorsiva è piuttosto immediato. Essa però deve essere riformulata in modo che non sia più ricorsiva, altrimenti il costo asintotico non può essere quantificato, e questa è la parte meno semplice.
- La riformulazione della funzione di costo ricorsiva in una equivalente ma non ricorsiva si affronta impostando una *equazione di ricorrenza*, costituita dalla **formulazione ricorsiva e dal caso base.**

Equazioni di ricorrenza (2)

Esempio: costo computaz. della ricerca sequenziale ricorsiva:

```
Funzione Ricerca_seq_ric(A: vettore; v: intero; n: intero)   T(n)
    if (A[n] = v)                                            Θ(1)
        return true
    else
        if (n = 1)                                           Θ(1)
            return false
        else
            return Ricerca_seq_ric (A, v, n - 1)             T(n-1)
```

In generale: $T(n)=\Theta(1)+T(n-1)$

Caso base: $T(1)=\Theta(1)$

Equazioni di ricorrenza (3)

- La parte generale dell'equazione di ricorrenza che definisce *T(n)* deve essere sempre costituita dalla somma di almeno due addendi, di cui almeno uno contiene la parte ricorsiva (nell'esempio *T(n-1)*) mentre uno rappresenta il costo computazionale di tutto ciò che viene eseguito al di fuori della chiamata ricorsiva.
- Anche se questa parte dovesse essere un solo confronto, il suo costo non può essere ignorato, altrimenti si otterrebbe che la funzione ha una costo computazionale indipendente dalla dimensione del suo input, e questa è data da quanto illustrato nel caso base. Non possiamo dire che ciò sia impossibile, ma è molto improbabile.
- Deve sempre essere presente un caso base.

Equazioni di ricorrenza (4)

Esistono alcuni metodi utili per risolvere le equazioni di ricorrenza, che illustreremo:

- metodo di sostituzione
- metodo iterativo
- metodo dell'albero
- (metodo principale)

Metodo di sostituzione (1)

Idea:

- si ipotizza una soluzione per l'equazione di ricorrenza data;
- si verifica se essa "funziona".

Difficoltà:

- si deve trovare la funzione più vicina alla vera soluzione, perché tutte le funzioni più grandi (se stiamo cercando O) o più piccole (se stiamo cercando Ω) funzionano.
- In effetti questo metodo serve soprattutto nelle dimostrazioni mentre si sconsiglia di utilizzarlo nella pratica.

Metodo di sostituzione (2)

Esempio: Equazione relativa alla ricerca seq. ricorsiva

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

- Ipotizziamo la soluzione $T(n)=cn$ per una costante opportuna c.
- $T(n)=cn=c(n-1)+ \Theta(1)$, cioè $c= \Theta(1)$;
- $T(1)=c= \Theta(1)$,
- Non è detto che le due costanti nascoste nella notazione asintotica siano le stesse. Per questa ragione, è necessario eliminare la notazione asintotica dall'equazione di ricorrenza, così che le costanti non rimangano "nascoste" nella notazione, conducendo a risultati errati
- ...

Metodo di sostituzione (3)

segue Esempio: Equazione relativa alla ricerca seq. ricorsiva

- $T(n) = T(n - 1) + c$
- $T(1) = d$

per due costanti c e d fissate.

- Ipotizziamo la soluzione $T(n) = O(n)$, ossia $T(n) \leq kn$ dove k è una costante che va ancora determinata.
- N.B. non si può sperare in una soluzione esatta, ma possiamo solo maggiorare o minorare.
- Sostituiamo nel caso base:

$T(1) \leq k$; poiché sapevamo che $T(1) = d$, la disuguaglianza è soddisfatta se e solo se $k \geq d$.

- Sostituiamo nella formulazione ricorsiva:

$T(n) \leq k(n-1) + c = kn - k + c \leq kn$; vera se e solo se $k \geq c$.

- La soluzione $T(n) \leq kn$ è corretta per tutti i valori di k tali che $k \geq c$ e $k \geq d$. Poiché un tale k esiste sempre, una volta fissati c e d, $T(n)$ è un $O(n)$.

Metodo di sostituzione (4)

segue Esempio: Equazione relativa alla ricerca seq. ricorsiva

- $T(n) = T(n - 1) + c$
- $T(1) = d$

per due costanti c e d fissate.

- Che cosa sarebbe successo se avessimo ipotizzato $T(n) \leq kn^2$?
- Anche questa soluzione è corretta per tutti i valori di k tali che $k \geq c$ e $k \geq d$.
- A noi interessa stimare $T(n)$ asintoticamente tramite la funzione più piccola, e questo è spesso un obiettivo difficile.

Metodo di sostituzione (5)

segue Esempio: Equazione relativa alla ricerca seq. ricorsiva

- $T(n) = T(n - 1) + c$
- $T(1) = d$

per due costanti c e d fissate.

- Per rendere il nostro risultato stretto, ipotizziamo ora la soluzione $T(n) = \Omega(n)$, ossia $T(n) \geq hn$ dove h è una costante che va ancora determinata.
- In modo assolutamente analogo al caso O, sostituiamo nel caso base ottenendo $h \leq d$,
- Sostituiamo nella formulazione ricorsiva dell'equazione di ricorrenza ottenendo $T(n) \geq h(n - 1) + c = hn - h + c \geq hn$ vera se e solo se $h \leq c$.
- Deduciamo dunque che $T(n)$ è un $\Omega(n)$ poiché è possibile trovare un valore h ($h \leq c$, $h \leq d$) per il quale si ha $T(n) \geq hn$.

Metodo di sostituzione (6)

segue Esempio: Equazione relativa alla ricerca seq. ricorsiva

- $T(n) = T(n - 1) + c$
- $T(1) = d$

per due costanti *c* e *d* fissate.

Dalle due soluzioni: $T(n) = O(n)$ e $T(n) = \Omega(n)$, si ottiene ovviamente che $T(n) = \Theta(n)$.

Metodo iterativo (1)

UNITELMA SAPIENZA
SAPIENZA
UNIVERSITÀ DI ROMA
DIPARTIMENTO DI INFORMATICA

Idea:

- sviluppare l'equazione di ricorrenza ed esprimerla come somma di termini dipendenti da *n* e dal caso base

Difficoltà:

- maggiore quantità di calcoli algebrici rispetto al metodo di sostituzione

Metodo iterativo (2)

Esempio: Equazione relativa alla ricerca seq. ricorsiva

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

$T(n) = T(n - 1) + \Theta(1)$, ma $T(n\text{-}1)=T(n\text{-}2) + \Theta(1)$

Sostituiamo:

$T(n) = T(n - 2) + \Theta(1) + \Theta(1)$, ma $T(n\text{-}2)=T(n\text{-}3) + \Theta(1)$

$T(n) = T(n - 3) + \Theta(1) + \Theta(1) + \Theta(1)$, ma $T(n\text{-}3)=T(n\text{-}4) + \Theta(1)$

$T(n) = T(n - 4) + \Theta(1) + \Theta(1) + \Theta(1) + \Theta(1)$, ecc.

fino a ottenere:

$T(n) = n\,\Theta(1) = \Theta(n)$.

Metodo iterativo (3)

Esempio: Equazione relativa alla ricerca binaria ricorsiva

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

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

Sostituiamo:

$T(n)= T(n/2^2) + \Theta(1)+ \Theta(1)$ ma $T(n/2^2)= T(n/2^3) + \Theta(1)$

$T(n)= T(n/2^3) + \Theta(1)+ \Theta(1) + \Theta(1)$ ma…

…

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

Quando $k = \log n$ si ha $n/2^k = 1$, quindi per tale valore di k ci fermiamo ed otteniamo:

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

Metodo iterativo (4)

Esempio: Equazione relativa al calcolo di Fib(n)

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

Non si riesce a risolvere questa equazione di ricorrenza tramite il metodo iterativo, in quanto il numero di addendi cresce esponenzialmente giungendo presto ad una quantità non gestibile.

Ma:

- $T(n) \leq 2T(n-1) + \Theta(1)$
  che ci permetterà di derivare un limite superiore O;
- $T(n) \geq 2T(n-2) + \Theta(1)$
  che ci permetterà di derivare un limite inferiore Ω;

Metodo iterativo (5)

segue Esempio: Equazione relativa al calcolo di Fib(n)

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

Limite superiore:

$$T(n) \leq 2T(n-1) + \Theta(1) \leq 2^2T(n-2) + 2^1\Theta(1) + \Theta(1) \leq$$

$$\leq 2^3T(n-3) + 2^2\Theta(1) + 2^1\Theta(1) + \Theta(1) \leq \ldots \leq$$

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

Il procedimento si ferma per $n - k = 1$, ossia $k = n - 1$ per cui:

$$T(n) \leq 2^{n-1}\,\Theta(1) + \sum_{i=0}^{n-2} 2^i\Theta(1) = 2^{n-1}\,\Theta(1) + (2^{n-1} - 1)\Theta(1) =$$

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

Segue che $T(n) = O(2^n)$

Metodo iterativo (6)

segue Esempio: Equazione relativa al calcolo di Fib(n)

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

Limite inferiore:

$T(n) \geq 2T(n-2) + \Theta(1) \geq 2^2T(n-4) + 2^1\Theta(1) + \Theta(1) \geq$

$\geq 2^3T(n-6) + 2^2\Theta(1) + 2^1\Theta(1) + \Theta(1) \geq \ldots \geq$

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

Il procedimento si ferma per $n - 2k = 1$, ossia $k = n/2$ per cui:

$T(n) \geq 2^{n/2}\,\Theta(1) + \sum_{i=0}^{n/2-1} 2^i\Theta(1) = 2^{n/2}\,\Theta(1) + (2^{n/2} - 1)\Theta(1) =$

$=(2 * 2^{n/2} - 1)\,\Theta(1)$

Segue che $T(n) = \Omega(2^{n/2})$

Metodo iterativo (7)

segue Esempio: Equazione relativa al calcolo di Fib(n)

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

N.B. anche se non siamo in grado di trovare una funzione asintotica precisa (Θ), possiamo comunque concludere che il calcolo dei numeri di Fibonacci con una tecnica ricorsiva richiede un costo computazionale esponenziale in n, visto che

$k_1 2^{n/2} \leq T(n) \leq k_2 2^n$

per opportune costanti k_1 e k_2.

Metodo dell'albero (1)

Idea:

- rappresentare graficamente lo sviluppo del costo computazionale dell'algoritmo, in modo da poterla valutare con una certa facilità

Difficoltà:

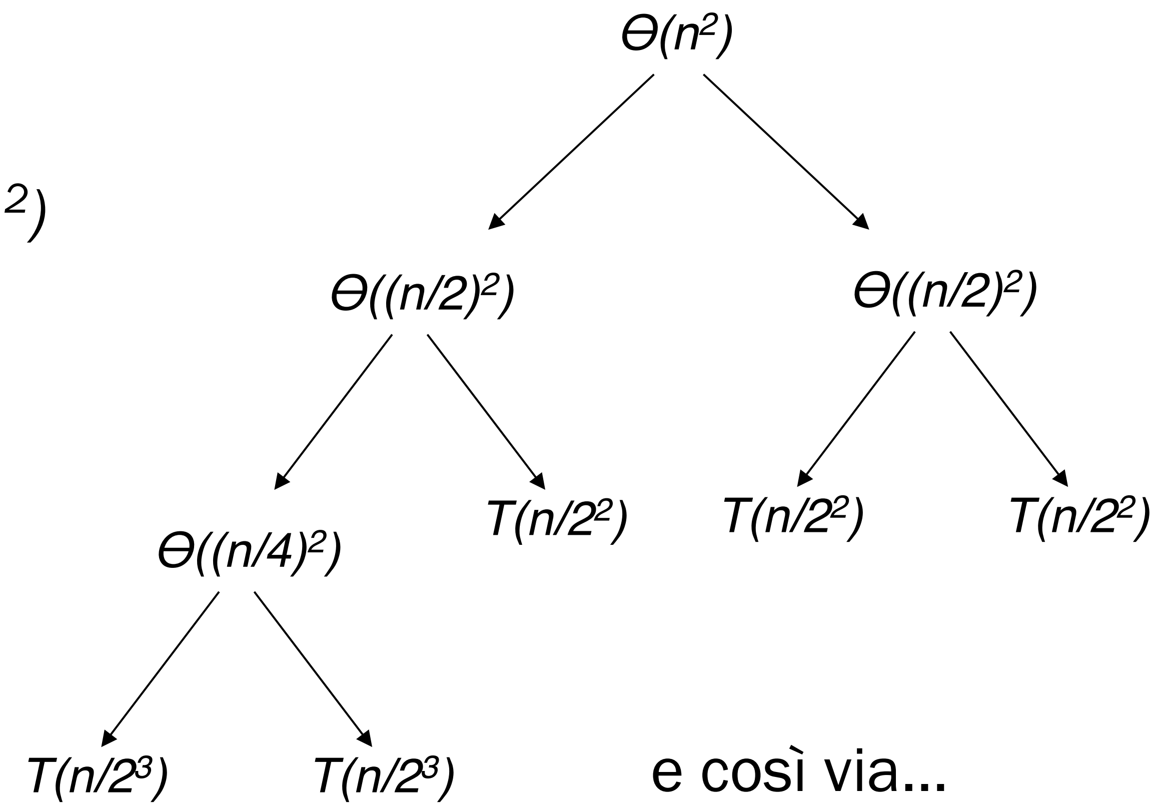
- come il metodo iterativo

Metodo dell'albero (2)

Esempio:

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

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

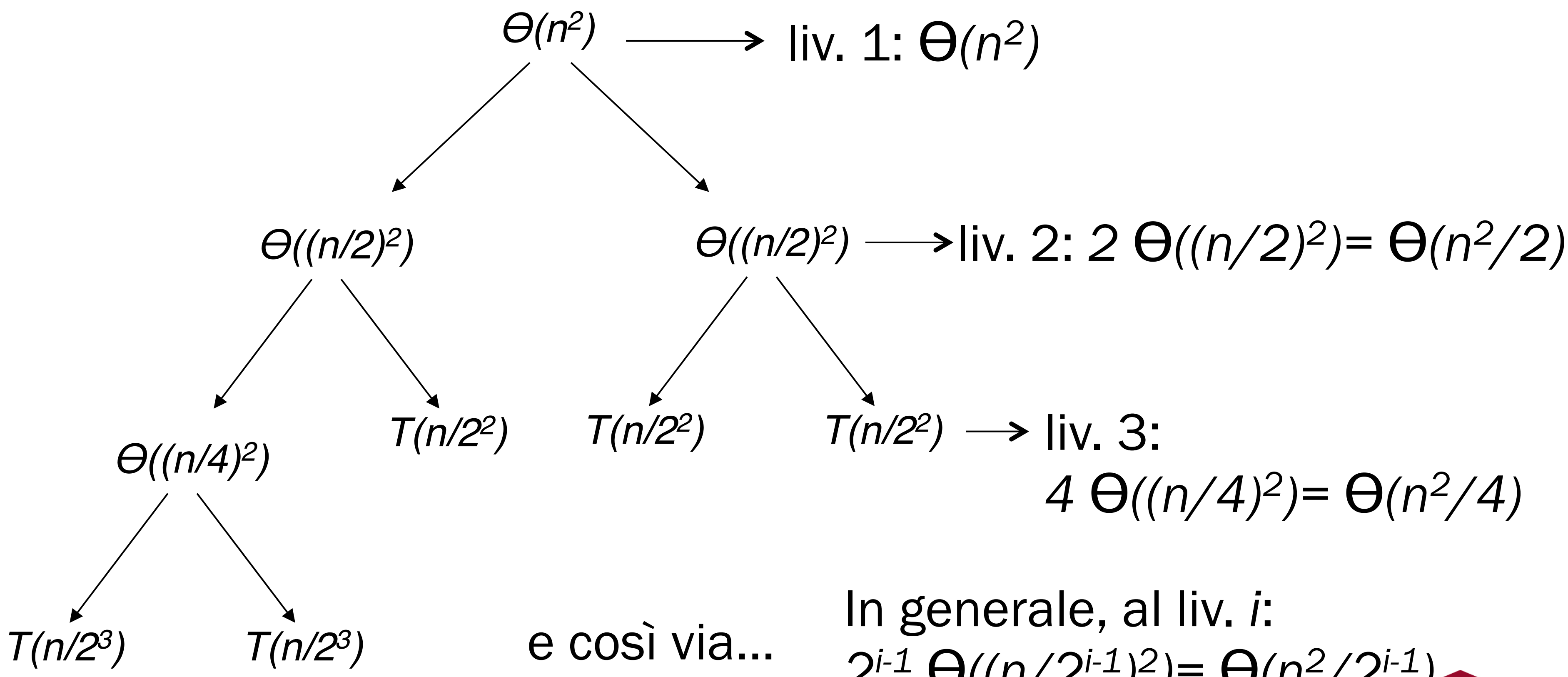


e così via…

Metodo dell'albero (3)

segue Esempio:

Una volta completato l'albero, il costo computazionale è dato dalla somma dei contributi di tutti i livelli (cioè le "righe" in cui sono disposti i nodi) di cui è costituito l'albero.



In generale, al liv. *i*:

$2^{i-1}\,\Theta((n/2^{i-1})^2)=\Theta(n^2/2^{i-1})$

Metodo dell'albero (4)

segue Esempio:

In questo caso il numero di livelli dell'albero ha un valore tale che $n/2^{i-1}=1$, ossia $i-1=\log n$ da cui $i=\log n+1$.

Sommiamo i contributi di tutti i livelli:

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

Calcolare la soluzione delle seguenti equazioni di ricorrenza con tutti e tre i metodi:

- $T(n)=2T(n/2)+\Theta(n)$ $\quad T(1)=\Theta(1)$
- $T(n)=3T(n/2)+\Theta(n)$ $\quad T(1)=\Theta(1)$
- $T(n)=3T(n/4)+\Theta(n)$ $\quad T(1)=\Theta(1)$
- $T(n)=2T(n/2)+\Theta(n)$ $\quad T(1)=\Theta(1)$
- $T(n)=4T(n/2)+\Theta(n)$ $\quad T(1)=\Theta(1)$
- $T(n)=2T(n/2)+\Theta(n^3)$ $\quad T(1)=\Theta(1)$
- $T(n)=16T(n/4)+\Theta(n^2)$ $\quad T(1)=\Theta(1)$
- $T(n)=T(n-1)+\Theta(n)$ $\quad T(1)=\Theta(1)$
- $T(n)=3T(n/2)+\Theta(n \log n)$ $\quad T(1)=\Theta(1)$