

In questa lezione

- **Esempi di relazioni di ricorrenza**
- **[CLRS] cap 4, par. 4.1 e 4.2**

Esempio albero e sviluppo

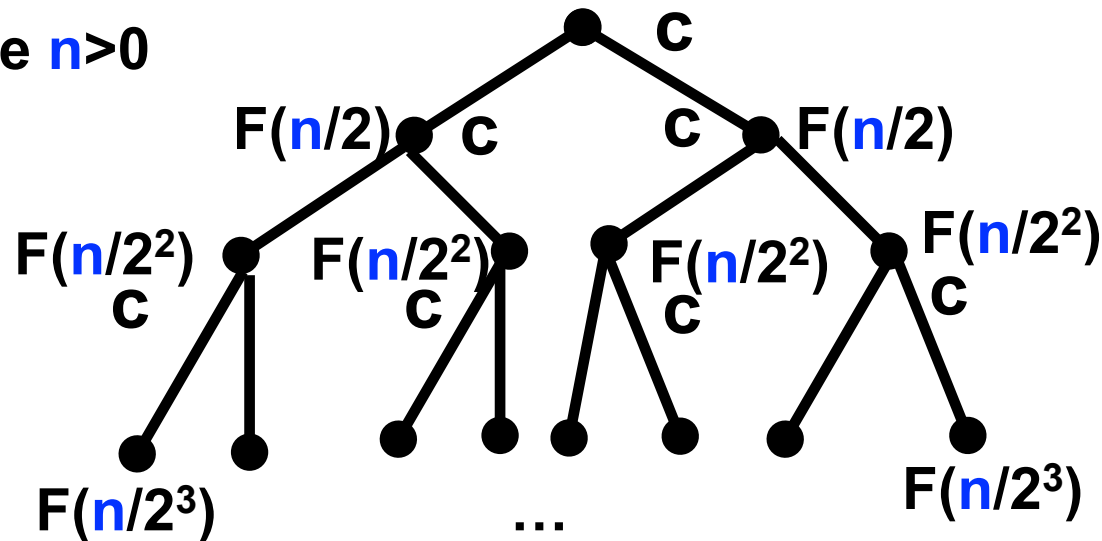
$$F(n) = \begin{cases} 1 & \text{se } n=1 \\ 2F(n/2) + c & \text{se } n>0 \end{cases}$$

$$\begin{aligned} F(n) &= 2F(n/2) + c \\ &= 2(2F(n/2^2) + c) + c \\ &= 2^2F(n/2^2) + 2c + c \\ &= 2^2(2F(n/2^3) + c) + 2c + c \\ &= 2^3F(n/2^3) + 2^2c + 2c + c \end{aligned}$$

Se $n=2^h$ si ha

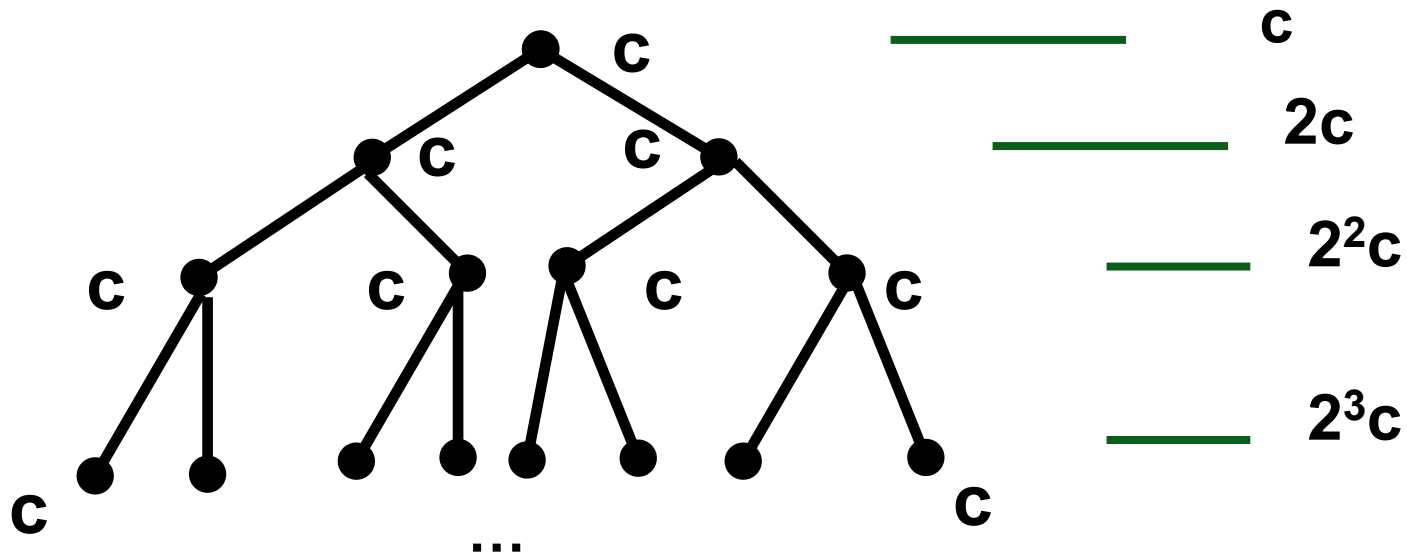
$$F(n) = 2^h F(1) + 2^{h-1}c + \dots + 2c + c = n + c(2^h - 1) = n + c(n - 1).$$

Quindi supponiamo che $F(n) = \Theta(n)$



Uso dell'albero

$$F(n) = \begin{cases} 1 & \text{se } n=1 \\ 2F(n/2) + c & \text{se } n>0 \end{cases}$$



L'albero ha altezza h e l'ultimo livello ha 2^h foglie di "costo" 1.

Ogni livello i ha "costo" totale $2^i c$ quindi

$$F(n) = 2^h 1 + 2^{h-1}c + \dots + 2c + c = n + c(2^h - 1) = n + c(n - 1).$$

Esempio 2

La ricorrenza

$$T(n) = d \text{ per } n \leq 1$$

$$T(n) = T(n/3) + \Theta(n^2) \text{ altrimenti}$$

viene riscritta:

$$T(n) = d \text{ per } n \leq 1$$

$$T(n) = T(n/3) + cn^2$$

Esempio 2

$$T(n) = d \text{ per } n \leq 1$$

$$T(n) = T(n/3) + cn^2$$

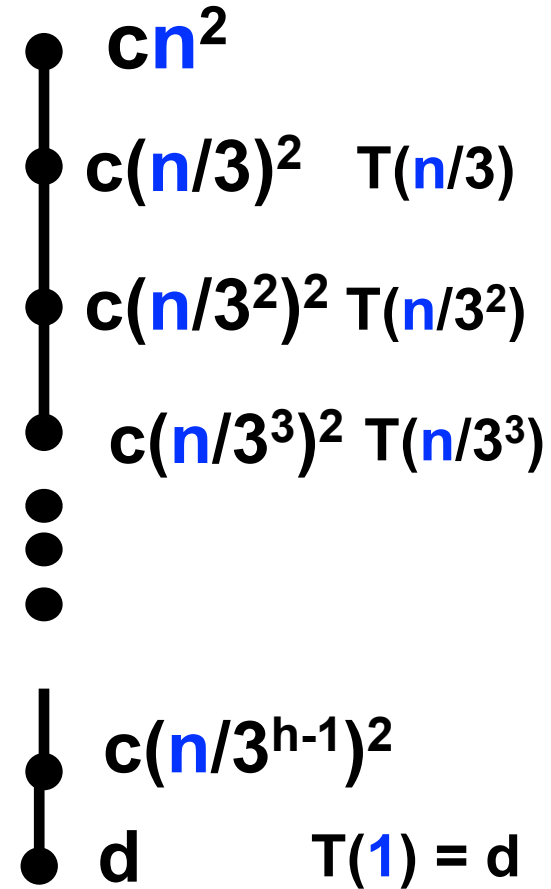
$$T(n) = T(n/3) + cn^2$$

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

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

...

$$n/3^h = 1$$



Fase 1 - i conteggi

$$T(n) = d \text{ per } n \leq 1$$

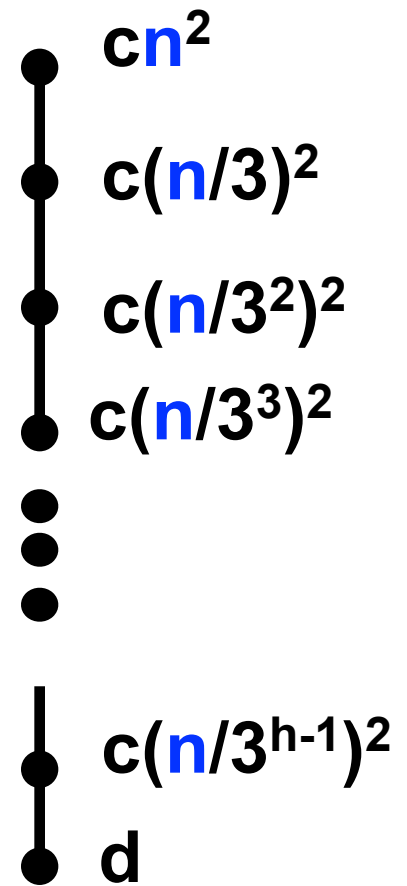
$$T(n) = T(n/3) + cn^2$$

$$T(n) =$$

$$cn^2 + (1/3)^2 cn^2 + (1/3^2)^2 cn^2 + (1/3^3)^2 cn^2 \\ \dots + (1/3^{h-1})^2 cn^2 + d =$$

$$cn^2 [((1/3)^2)^0 + (1/3)^2 + (1/3^2)^2 + (1/3^2)^3 \dots \\ + (1/3^2)^{h-1}] + d =$$

$O(n^2)$ perché la serie potenze di base $1/9$, che è minore di 1, converge a una costante. Poiché la serie maggiore la somma parziale da 0 a $h-1$, otteniamo il risultato



Verifica induttiva

$$T(n) = d \text{ per } n \leq 1$$

$$T(n) = T(n/3) + cn^2$$

1. Vogliamo far vedere che $T(n) = O(n^2)$ cioè che esistono due costanti $b, n_0 \geq 0$ tali che $T(n) \leq bn^2$, per ogni $n \geq n_0$

Passo induttivo:

Supponiamo che sia vero per ogni $m < n$ e consideriamo $T(n)$

$$T(n) = T(n/3) + cn^2 \text{ per l'ipotesi induttiva}$$

$$T(n/3) \leq b(n/3)^2 \text{ quindi}$$

$$T(n) \leq b(n/3)^2 + cn^2$$

Bisogna controllare se è vero che $b(n/3)^2 + cn^2 \leq bn^2$

$$\Leftrightarrow b(1/9)n^2 + cn^2 \leq bn^2 \text{ se } n \geq 1, \text{ possiamo dividere per } n^2$$

$$\Leftrightarrow 1/9b + c \leq b$$

$$\Leftrightarrow b + 9c \leq 9b$$

$$\Leftrightarrow 9c \leq 8b \text{ quindi basta prendere per esempio } b = 9c,$$

$$\text{e } n_0 = 1.$$

Verifica di $T(1)$ e limite inferiore

$$T(n) = d \text{ per } n \leq 1$$

$$T(n) = T(n/3) + cn^2$$

Abbiamo trovato che

$$T(n) \leq bn^2 \text{ per } b = 9c \text{ e ogni } n \geq 1,$$

$$\text{ma } T(1) = d$$

Allora prendiamo $b = 9(c+d)$, così possiamo lasciare $n_0 = 1$.

Poiché il termine cn^2 fornisce un limite inferiore alla crescita della funzione T , possiamo concludere che $T(n) = \theta(n^2)$

Ricerca binaria: versione ricorsiva

INPUT: un array A di n numeri con indice iniziale 1, e l'elemento da cercare, x .

Precondizione: gli elementi in A sono **ordinati in ordine crescente**

OUTPUT: la posizione di x se occorre nella sequenza, 0 altrimenti.

```
RicBinariaRic(A, low, high, x)
```

```
if (low > high) then return 0
```

```
mid ← (high + low + 1)/2
```

```
if x = A[mid] then return mid
```

```
else if (x < A[mid])
```

```
    then return RicBinariaRic(A, low, mid-1, x)           T(n/2)
```

```
        /*x può essere nella metà sinistra*/
```

```
    else return RicBinariaRic(A, mid+1, high, x)         T(n/2)
```

```
        /*x può essere nella metà destra*/
```

} $\Theta(1)$

L'intervallo di ricerca è sempre tra $A[\text{low}]$ e $A[\text{high}]$ compreso

La ricorrenza

$$T(n) \leq T(n/2) + \Theta(1)$$

Esplicitiamo le costanti:

$$T(n) \leq T(n/2) + c \text{ se } n \geq 1$$

$$T(1) = d \text{ altrimenti}$$

**Dimostriamo che $T(n) = O(\lg n)$,
cioè che esistono due costanti $k, n_0 \geq 0$
tali che $T(n) \leq k \lg n$, per ogni $n \geq n_0$**

$O(\lg n)$ per Ricerca Binaria

Passo induttivo: supponiamo che $T(m) \leq k \lg m$ per ogni $m < n$ e consideriamo $T(n)$:

$$T(n) \leq T(n/2) + c$$

$$\leq k \lg n/2 + c$$

$$\leq k \lg n - k + c$$

$$\leq k \lg n \text{ per ogni } k \geq c \text{ e } n \geq 2,$$

perché per $n = 1 \lg n = 0$

Controlliamo il caso base $T(1) = d$. Per definizione $T(2) = T(1) + c = d + c$.

Prendendo $k = d+c$, possiamo dire che $T(2) \leq k \lg 2 = c+d$.
Quindi concludiamo che $T(n) \leq k \lg n$ per ogni $n \geq 2$.

Semplificare la ricorrenza nell'analisi

Poiché il caso base, nel contesto dell'analisi degli algoritmi, è sempre una costante, possiamo ometterlo.

Per esempio per descrivere la complessità del mergesort con una ricorrenza possiamo scrivere

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

invece di

$$\begin{aligned} T(n) &= \Theta(1) \text{ se } n \leq 1 \\ T(n) &= 2T(n/2) + \Theta(n) \text{ altrimenti} \end{aligned}$$

Esempio 3

La ricorrenza

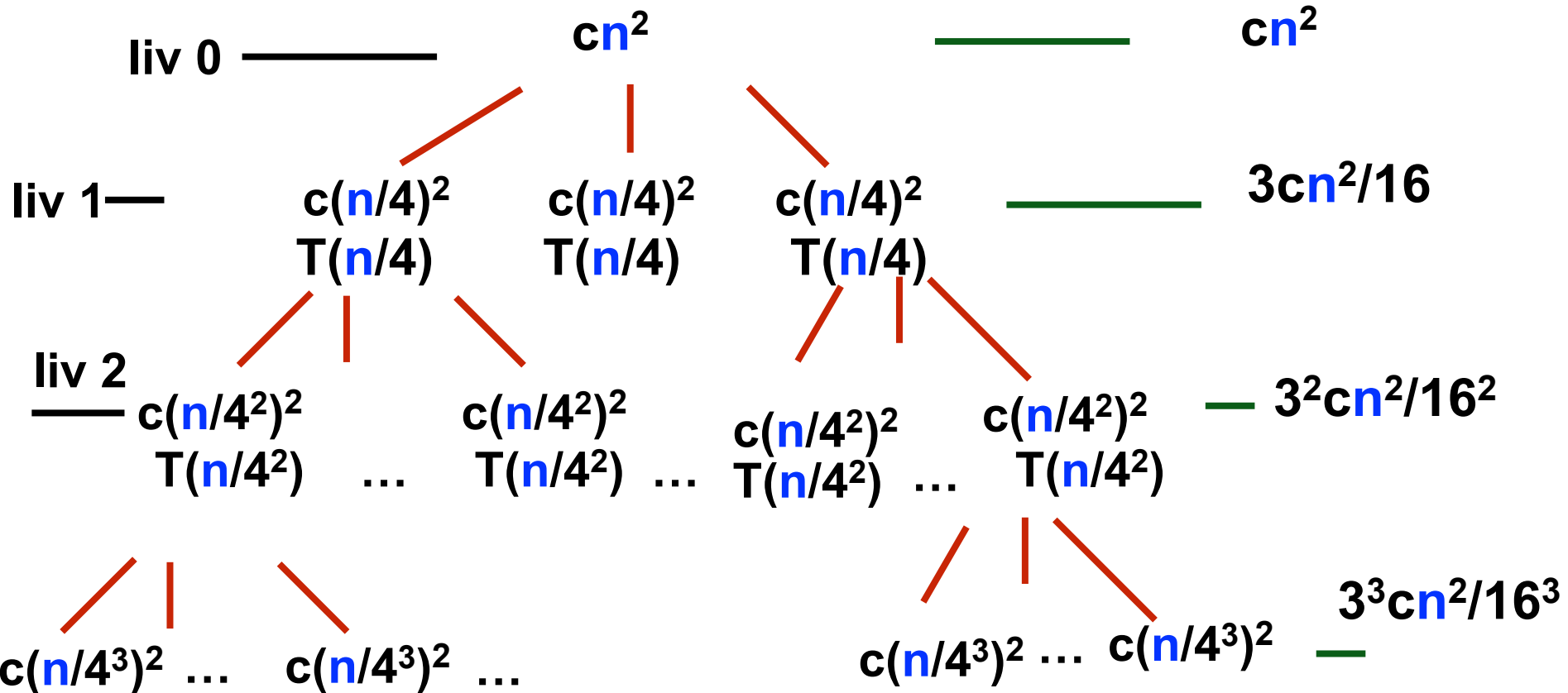
$$T(n) = 3T(n/4) + \Theta(n^2)$$

Albero di ricorsione

$$T(n) = c \text{ se } n \leq 1$$

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

Sia $n = 4^h$

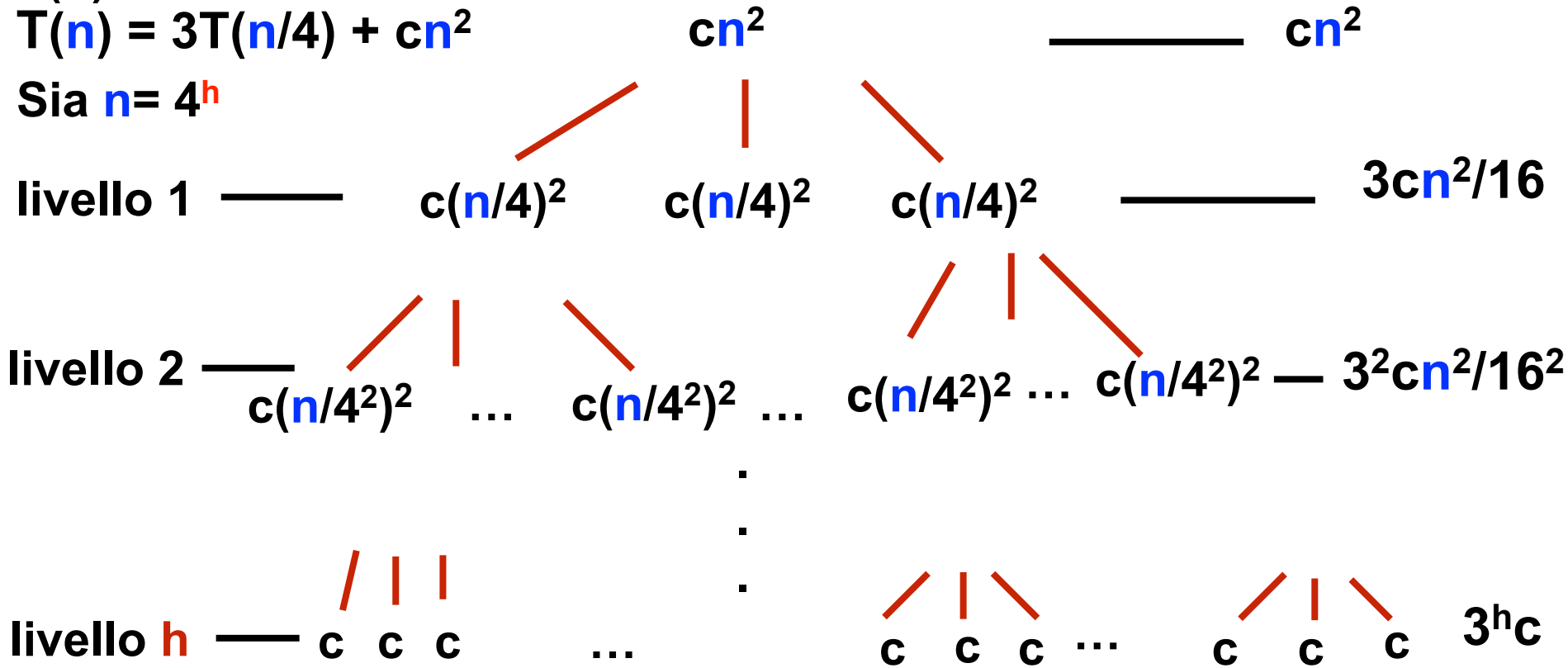


Albero di ricorsione

$$T(n) = c \text{ se } n \leq 1$$

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

$$\text{Sia } n = 4^h$$



$$h = \log_4 n, \text{ con un cambio di base } \lg_4 n = \log_3 n \log_4 3$$

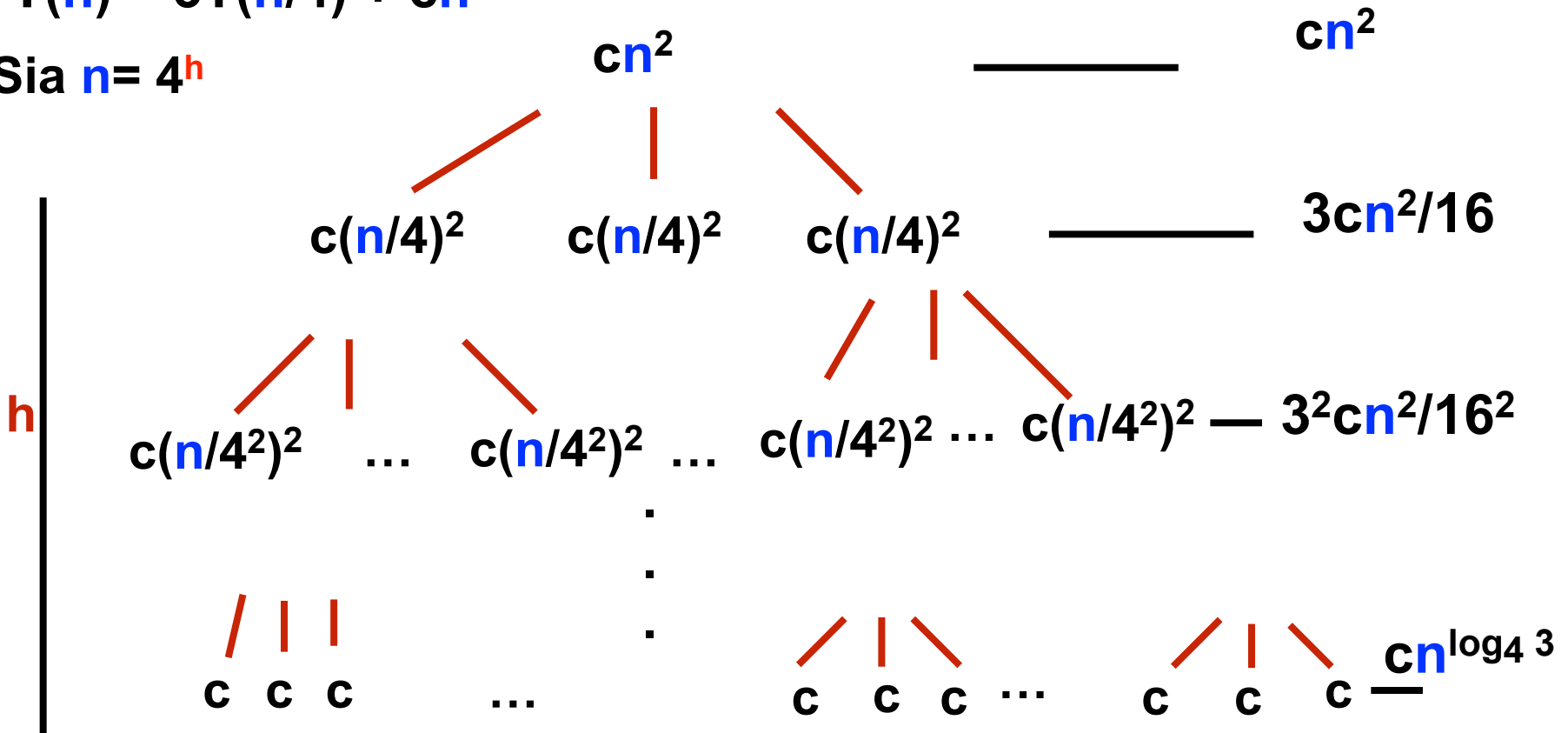
$$3^h c = 3^{\log_3 n \log_4 3} c = n^{\log_4 3} c$$

Albero di ricorsione

$$T(n) = c \text{ se } n \leq 1$$

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

Sia $n = 4^h$



$$T(n) = (3/16)^0 cn^2 + (3/16)cn^2 + (3/16)^2 cn^2 + \dots + (3/16)^{h-1} cn^2 + cn^{\log_4 3}$$

Conteggi

$T(n) =$

$$(3/16)^0 cn^2 + (3/16)cn^2 + (3/16)^2 cn^2 + \dots + (3/16)^{h-1} cn^2 + cn^{\log_4 3}$$

$$= cn^2 [(3/16)^0 + (3/16) + (3/16)^2 + \dots + (3/16)^{h-1}] + cn^{\log_4 3}$$

Poiché $3/16 < 1$, se maggioriamo la somma di potenze finita con la somma estesa all'infinito, possiamo sfruttare il fatto che questa serie converge a una costante

$$\sum_{i \geq 0} (3/16)^i = \Theta(1)$$

Potremmo concludere che $T(n) = O(n^2)$, visto che $\log_4 3 < 2$

Siamo sicuri di questo risultato?

Fase 2: verifica

$$T(n) = 3T(n/4) + \Theta(n^2)$$

1. Vogliamo far vedere che $T(n) = O(n^2)$ cioè che esistono d ed n_0 tali che $T(n) \leq dn^2$, per ogni $n \geq n_0$.

Usiamo l'induzione.

Passo induttivo:

Supponiamo che sia vero per ogni $m < n$ e consideriamo $T(n)$

$$T(n) = 3T(n/4) + cn^2 \text{ poiché per ip. induttiva } T(n/4) \leq d(n/4)^2$$

$$3T(n/4) + cn^2 \leq 3d(n/4)^2 + cn^2 = (3/16)dn^2 + cn^2$$

Bisogna controllare se è vero che $(3/16)dn^2 + cn^2 \leq dn^2$

$$\Leftrightarrow 3/16d + c \leq d$$

$$\Leftrightarrow 3d + 16c \leq 16d$$

$$\Leftrightarrow 16c \leq 13d$$

$$\Leftrightarrow 16/13c \leq d$$

Vero se si prende $d \geq (16/13)c$ e $n_0 = 1$

verifica: passo base

$$T(n) = c \text{ se } n \leq 1$$

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

Abbiamo ottenuto $T(n) \leq dn^2$, per $d = 16/13c$ e per ogni $n \geq 1$.

Poiché $T(1) = c$, e $c \leq 16/13c$ si può anche mantenere $n_0 = 1$.

Conclusione

$$T(n) = c \text{ se } n \leq 1$$

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

Abbiamo provato che $T(n) = O(n^2)$.

Osserviamo che il termine additivo è in $\Theta(n^2)$, che fornisce un limite inferiore alla soluzione per $T(n)$.

Quindi possiamo dire che $T(n) = \Omega(n^2)$ e in definitiva che $T(n) = \Theta(n^2)$

Divide et impera: analisi

Dato un problema di dimensione (taglia) n

- **Dividi:** dividi il problema in a sottoproblemi ciascuno di taglia n/b $D(n)$
- **Conquista:** risolvi ricorsivamente i sottoproblemi, $aT(n/b)$
 - quando il sottoproblema è di taglia minore o uguale a c risolvilolo direttamente $\Theta(1)$
- **Combina:** usa le soluzioni dei sottoproblemi per determinare la soluzione del problema dato $C(n)$

$$T(n) = \Theta(1) \quad \text{se } n \leq c$$

$$T(n) = aT(n/b) + D(n) + C(n) \quad \text{se } n > c$$

Le ricorrenze da analisi di programmi ricorsivi

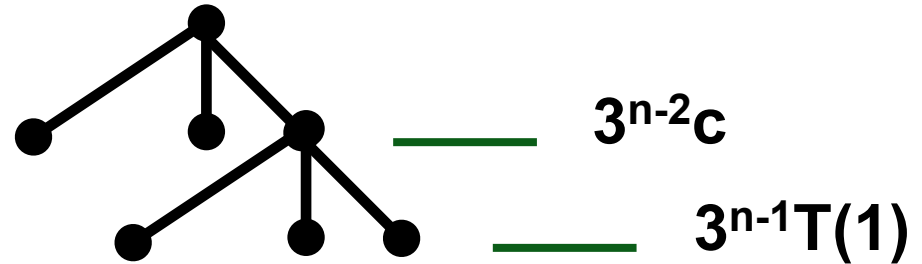
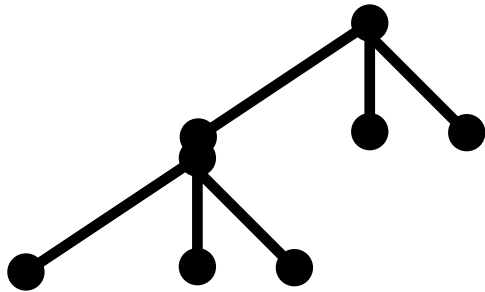
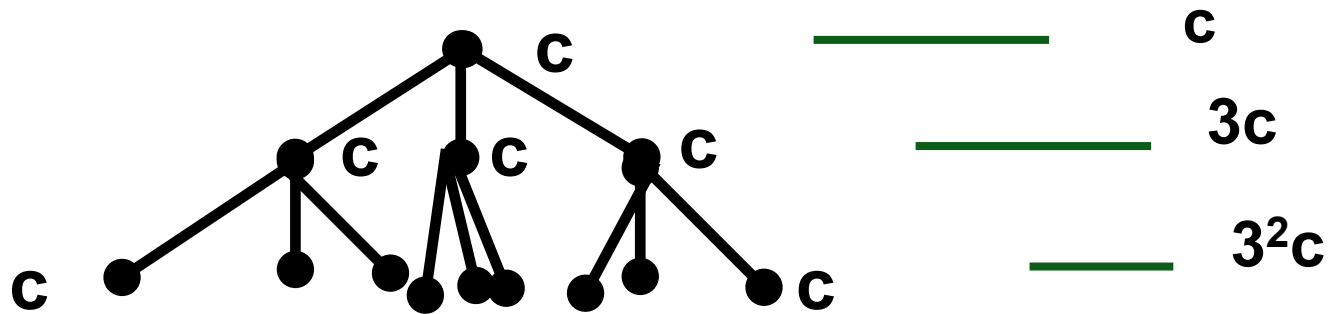
Si valuti la complessità del seguente algoritmo ricorsivo,

```
test(n)
  if  $n \leq 1$  then return
  for  $i \leftarrow 1$  to 3 do
    test(n-1)
```

$$T(n) = 3T(n-1) + \Theta(1)$$

Soluzione

$$T(n) = \begin{cases} \Theta(1) & \text{se } n = 1 \\ 3T(n-1) + c & \text{se } n > 1 \end{cases}$$



$$T(n) = 3^{n-1}T(1) + c \sum_{i=0, \dots, n-2} 3^i$$
$$= 3^{n-1}T(1) + \Theta(3^{n-1}) = \Theta(3^n)$$

Verifica induttiva

$$T(n) = \begin{cases} \Theta(1) & \text{se } n = 1 \\ 3T(n-1) + c & \text{se } n > 1 \end{cases}$$

1. Vogliamo far vedere che $T(n) = O(3^n)$ cioè che esistono due costanti $b, n_0 \geq 0$ tali che $T(n) \leq b3^n$, per ogni $n \geq n_0$

Passo induttivo:

Supponiamo che sia vero per ogni $m < n$ e consideriamo $T(n)$

$T(n) = 3T(n-1) + c$ per l'ipotesi induttiva

$T(n-1) \leq b3^{n-1}$ quindi

$T(n) \leq 3b3^{n-1} + c$

Bisogna controllare se è vero che $b3^n + c \leq b3^n$

$\Leftrightarrow b3^{n-1} + c \leq 3b3^{n-1}$

falso!

Ma il risultato è giusto, è solo un termine costante additivo che non fa tornare i conti. Facciamo vedere che esistono due costanti $b, n_0 \geq 0$ tali che $T(n) \leq b3^n - c$ per ogni $n \geq n_0$.

Verifica induttiva

$$T(n) = \begin{cases} \Theta(1) & \text{se } n = 1 \\ 3T(n-1) + c & \text{se } n > 1 \end{cases}$$

Facciamo vedere che esistono due costanti $b, n_0 \geq 0$ tali che $T(n) \leq b3^n - c$ per ogni $n \geq n_0$.

Passo induttivo:

Supponiamo che sia vero per ogni $m < n$ e consideriamo $T(n)$

$T(n) = 3T(n-1) + c$ per l'ipotesi induttiva

$T(n-1) \leq b3^{n-1} - c$ quindi

$T(n) \leq 3(b3^{n-1} - c) + c = b3^n - 3c + c = b3^n - 2c$

Bisogna controllare se è vero che $b3^n - 2c \leq b3^n$

Vero per ogni scelta di b !

Quindi esistono due costanti $b, n_0 \geq 0$ tali che $T(n) \leq b3^n - c$ per ogni $n \geq n_0$.

Trascuriamo il caso base.

Strange(A,i,j)

input A è un array e i e j due interi

prec. $1 \leq i, j \leq A.length$

if $i < j$ then

$m = (i+j)/2$

$m1 = (i+m)/2$ $\backslash\backslash$ m1 individua il punto

medio nella prima parte

$m2 = (m+j)/2$ $\backslash\backslash$ m2 quello della

seconda

Strange(A,m1,m2)

Strange (A,i,m)

Strange(A,m+1,j)

Final(A,i,j)

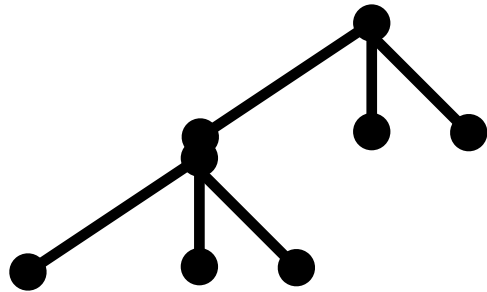
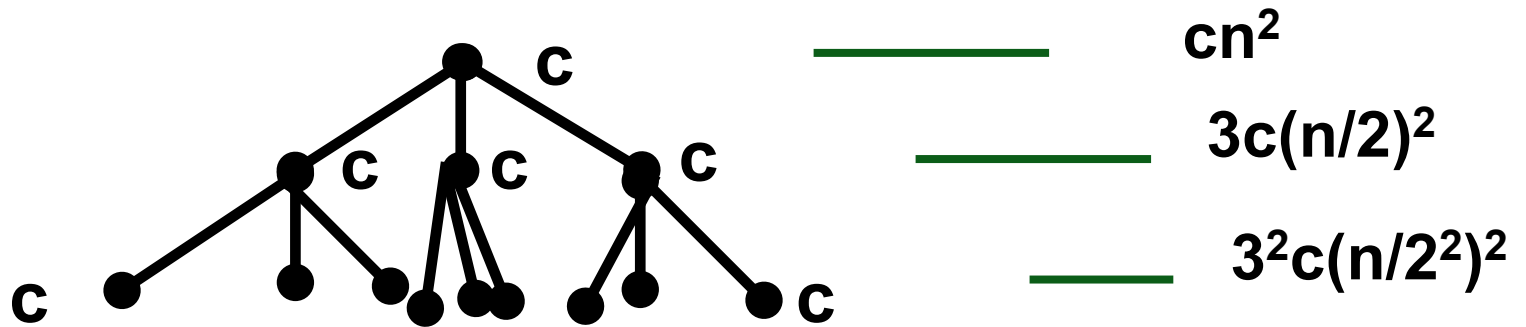
Final ha tempo di esecuzione $\Theta(n^2)$, dove n e' il numero degli elementi di A su cui è chiamata:

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

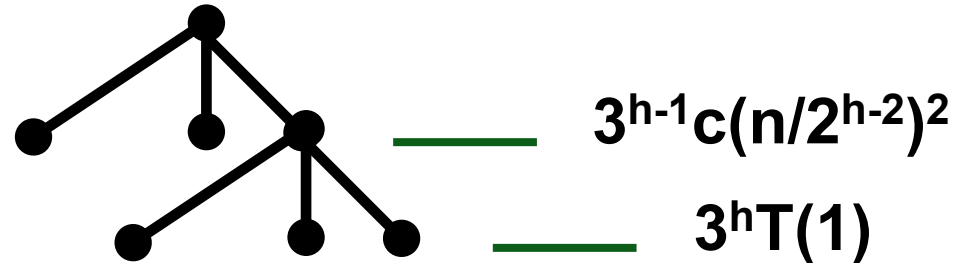
Soluzione

Sia $n = 2^h$

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



...



$$T(n) = 3^h T(1) + c n^2 \sum_{i=0, \dots, h-1} (3/2^2)^i = 3^h T(1) + c n^2 \Theta(1) \leq n^2 T(1) + \Theta(n^2) = \Theta(n^2)$$

Verifica induttiva

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

1. Vogliamo far vedere che $T(n) = O(n^2)$ cioè che esistono due costanti $b, n_0 \geq 0$ tali che $T(n) \leq bn^2$, per ogni $n \geq n_0$

Passo induttivo:

Supponiamo che sia vero per ogni $m < n$ e consideriamo $T(n)$

$T(n) = 3T(n/2) + cn^2$ per l'ipotesi induttiva si ha

$T(n/2) \leq b(n/2)^2$ quindi

$T(n) \leq 3b(n/2)^2 + cn^2$

Bisogna controllare se è vero che $3b(n/2)^2 + cn^2 \leq bn^2$

$$\Leftrightarrow 3/4 bn^2 + cn^2 \leq bn^2$$

$$\Leftrightarrow 3bn^2 + 4cn^2 \leq 4bn^2$$

$$\Leftrightarrow 4cn^2 \leq bn^2 \text{ se } n \geq 1$$

$$\Leftrightarrow 4c \leq b$$

Allora basta scegliere $b \geq 4c$ e $n_0 = 1$. Trascuriamo il caso base.

Esempio algoritmo ricorsivo

```
analizzami(int n)
c = 1
m = n
while (m > 1) :
    c++
    m=m/2
if n >1 then analizzami(n/2)
```

$\Theta(1)$

$\Theta(\lg n)$

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

Soluzione

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

Sia $n = 2^h$

$$\begin{aligned} T(n) &= T(n/2) + c \lg n = T(n/2^2) + c \lg n/2 + c \lg n \\ &= T(n/2^3) + c \lg n/2^2 + c \lg n/2 + c \lg n \end{aligned}$$

...

$$= T(1) + \sum_{i=0, \dots, h-1} c \lg n/2^i$$

$$= T(1) + c \sum_{i=0, \dots, h-1} (\lg n - i) =$$

$$T(1) + c \sum_{i=0, \dots, h-1} \lg n - c \sum_{i=0, \dots, h-1} i$$

$$\begin{aligned} &= T(1) + c h \lg n - c h(h-1)/2 . \text{ Ponendo } h = \lg n \\ &T(1) + c \lg^2 n - c/2 \lg^2 n - c \lg n/2 = \Theta(\lg^2 n) \end{aligned}$$

$O(\lg^2 n)$

Dimostriamo che $T(n) = O(\lg^2 n)$.

Ipotesi induttiva: supponiamo che $T(m) \leq k \lg^2 m$ per ogni $m < n$ e consideriamo $T(n)$:

$$T(n) = T(n/2) + c \lg n \leq k \lg^2 n/2 + c \lg n$$

per l'ipotesi induttiva

$$\leq k (\lg n - 1)^2 + c \lg n$$

$$\leq k (\lg^2 n + 1 - 2 \lg n) + c \lg n =$$

$$k \lg^2 n + k - 2 k \lg n + c \lg n$$

Si vuole che

$$k \lg^2 n + \underbrace{k - 2 k \lg n + c \lg n}_{\leq 0} \leq k \lg^2 n, \text{ cioè che}$$

$$k - 2 k \lg n + c \lg n \leq 0, \text{ cosa vera per esempio per } k = c \text{ e } n \geq 2$$

$\Omega(\lg^2 n)$

Dimostriamo che $T(n) = \Omega(\lg^2 n)$.

Ipotesi induttiva: supponiamo che $T(m) \geq k \lg^2 m$ per ogni $m < n$ e consideriamo $T(n)$:

$$\begin{aligned} T(n) &= T(n/2) + c \lg n \geq k \lg^2 n/2 + c \lg n \geq \\ &k (\lg n - 1)^2 + c \lg n \geq k(\lg^2 n + 1 - 2\lg n) + c \lg n \\ &= k \lg^2 n + k - 2k \lg n + c \lg n \end{aligned}$$

Si vuole che

$k \lg^2 n + k - 2k \lg n + c \lg n \geq k \lg^2 n$, cioè che

$k - 2k \lg n + c \lg n \geq 0$,

cosa vera per esempio per $k = c/2$ e $n \geq 1$

QuickSort: intuizione caso medio

Un caso che potrebbe prodursi:

$$T(n) = T(9n/10) + T(n/10) + cn$$

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

