

Tecniche per provare la correttezza / 01-03 / 07-03

1) Ricorsivi per induzione

- Insiemi definiti induttivamente

$\mathbb{N}) \quad 0 \in \mathbb{N}$

se $n \in \mathbb{N}$ allora $n+1 \in \mathbb{N}$

savibile anche

$$\frac{0}{0} \quad \frac{n}{n+1}$$

PARI) 0 e' pari

se $\underbrace{n}_{\text{e' pari}}$ allora $n+2$ e' pari $\frac{0}{0} \quad \frac{n}{n+2}$

(sfrutta la definizione per dimostrare: possi ricorsivi)

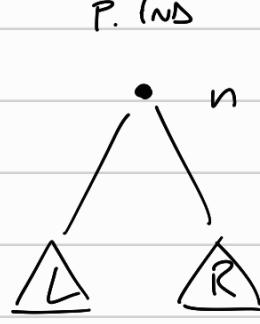
sequenze
/ liste

la lista vuota $\in L$ (e' uno isto) $\frac{}{} \quad \frac{\ell}{n:\ell}$
se $\ell \in L$, $n \in \mathbb{N}$ allora $n:\ell \in L$ E $\frac{}{} \quad \frac{n:\ell}{n:\ell}$

(un concatenato a ℓ , gli el. sono n nuovi)

alberi
binari

BASE
albero vuoto



(n naturale sono nodi)

compresso di foglie (figli vuoti)
e di quelli con figli da un lato

Se ho insiemi definiti inductivamente, posso provare proprietà dell'insieme in modo inductive.

Per trovare che vale

$$\frac{P(x) \quad \forall x \in X}{\text{PROPRIETÀ}}$$

basta far vedere
che ogni metaregola
(regola generale)

$$\frac{x_1, \dots, x_n}{x}$$

se vale $P(x_1), \dots, P(x_n)$
allora vale $P(x)$

base = regole
senza premesse

Esempi

$$\bullet \frac{n}{\sum_{n=0}^{n+1}}$$

$$\left[\begin{array}{l} P(0) \\ P(n) \rightarrow P(n+1) \end{array} \right] \rightarrow P(n) \quad \forall n \in \mathbb{N}$$

INDUZIONE

ARITMETICA

assumo di sapere che vale solo sui successivi

$$\bullet \frac{\{n \mid n < m\}}{m}$$

$$\left[\begin{array}{l} P(0) \\ P(n) \quad \forall n < m \rightarrow P(m) \end{array} \right] \rightarrow P(n) \quad \forall n \in \mathbb{N}$$

INDUZIONE FORTE (CORPIELA)

tipicamente utile per approcci
"divide et impera"

assumo di sapere che vale
su partite dai più piccoli

LISI

$$\bullet \frac{\ell}{\varepsilon \quad x:\ell}$$

$$\left| \begin{array}{l} P(\varepsilon) \\ P(\ell) \rightarrow P(x:\ell) \end{array} \right| \quad P(\ell) \quad \forall \ell \in \mathbb{N}^*$$

(insieme delle stringhe)

ALBERI

- albero vuoto



$$\begin{array}{c}
 P(\text{albero vuoto}) \\
 | \\
 P(L) \wedge P(R) \rightarrow \\
 \rightarrow P(\text{diagramma})
 \end{array}$$

$P(t)$ $\forall t$ albero binario

avremo vale anche per gli
alberi costituiti da quelli
di precedenza

In dettaglio

summa
di 1

$$P(n) \\ n \quad \sum_{i=1}^n i = \frac{n(n+1)}{2}$$

$\forall n \in \mathbb{N}$

IND. ARITMETICA

$$P(0) \quad 0 = 0 \quad \checkmark$$

la somma e' da 1 } e' equivalente,
ma $\forall n \in \mathbb{N}$ } l'induzione non
quindi parla da } cambia

Se vale $P(n)$ allora vale $P(n+1)$

hip induttiva

$$\sum_{i=1}^n i = \frac{n(n+1)}{2} \quad \text{vale per } n$$

Tesi
induttiva

$$\sum_{i=1}^{n+1} i = \frac{(n+1)(n+2)}{2} \quad \text{vale per } n+1$$

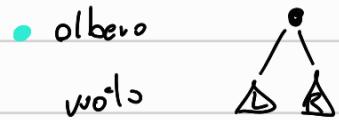
"

$$\sum_{i=1}^n i + (n+1) \quad \text{ultimo numero}$$

Dunque per l'ip. ind.

$$\frac{n(u_{i+1})}{z} - u_{i+1} = \frac{n(u_{i+1}) + z(u+2)}{z}$$

$$h_{i+1} \leq n^{\circ} \text{ nodi} \leq z^{h_{i+1}-1}$$



$$h(\Delta) = 1 + \max(h_L, h_R)$$

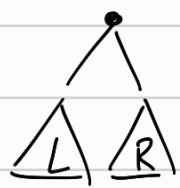
$$h(\text{albero vuoto}) = -1 \quad \text{per convenzione (e dopo l'ormai i conti)}$$

$$h(\bullet) = 0 \quad \text{ottico nodo isolato}$$

$$h(\Delta) = 1$$

$$P(\text{albero vuoto}) \quad 0 \leq 0 \leq 0 \quad \checkmark$$

Hip. induittiva : $P(L) \wedge P(R)$ Teo. : $P(\Delta)$



$$\begin{cases} h_{i+1} \leq n_L \leq z^{h_{i+1}} - 1 \\ h_{i+1} \leq n_R \leq z^{h_{i+1}} - 1 \end{cases} \quad \text{) le proprietà vale per i figli}$$

RADICE DEL
SOTTOALBERO R

n° nodi

Teo: $h_{i+1} \leq 1 + n_L + n_R \leq z^{h_{i+1}} - 1$) la proprietà vale per l'albero completo

assumo $h_R \geq h_L$ (equivalente l'altro modo)

$$\rightarrow h_{R+1} + 1 \leq 1 + n_L + n_R$$

\leq HP IND.

Keywords:

- Induzione
- " Aritmetica
- " Forte

09-03

$$h_{r+1} + 1 \leq 1 + n_L + n_R \leq 2^{h_{r+1}+1} - 1$$

$$h \leq n \leq 2^{h+1} - 1$$

$$\rightarrow 1 + n_L + n_R \leq 2^{h_{r+1}+1} - 1$$

$$\leq 2^1 \cdot 2^{h_{r+1}} - 1$$

$$1 + n_L + n_R \leq 2^{h_{r+1}} + (2^{h_{r+1}} - 1)$$

\leq per HP IND.

sappiamo $n_L \leq 2^{h_L+1} - 1$ e abbiamo assunto che $h_R \geq h_L$

quindi il secondo membro e' maggiore di h_r

$$n_L \leq 2^{h_L+1} - 1 \leq 2^{h_{r+1}} - 1$$

Uso induzione per provare la correttezza degli algoritmi ricorsivi

Approccio "divide et impere": $\text{alg}(P)$ $\dim(P) = n$

$$\dim(P_i) < \dim(P)$$



: if ($n \leq n_0$) caso base (soluzione diretta)
: else scomposizione dei problemi
 $\text{alg}(P_1) \dots \text{alg}(P_n)$
se \dim
setto una
soglia)
: composizione delle soluzioni

correttezza con induzione forte
(soluz. corrette per dim più
piccole)

2 mete'

"

(decomposizione costo costante)

Esempio :

- merge sort (divide in due il problema) → maggior lavoro nella composizione
- quick sort (dim dipende dal pivot) → maggior lavoro nella decomposizione

binary-search(x, a) // $a[0 \dots n-1]$ → sequenza ordinata

binary-search($x, a, 0, n-1$) // g ausiliarie



binary-search($x, a, \text{inf}, \text{sup}$)

if ($\text{inf} \leq \text{sup}$)

$$\text{mid} = (\text{inf} + \text{sup}) / 2$$

if ($x < a[\text{mid}]$) return binary-search($x, a, \text{inf}, \text{mid}-1$)

else if ($x > a[\text{mid}]$) return binary-search($x, a, \text{mid}+1, \text{sup}$)

else return true // $x = a[\text{mid}]$

else return false

Prove di correttezza di ind forte

$$n = \dim \text{array}$$

Base $n=0$ array vuoto, alg restituisce false ✓

P. Ind Assumo alg corretto $\forall m < n$

$$n > 0$$

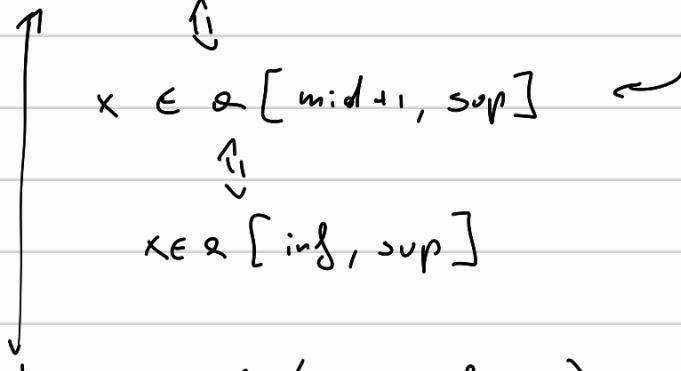
$\frac{\text{mid} = \text{inf} + \text{sup}}{2}$ Tre casi ↴

$\alpha[\text{mid}] = x$	$\text{inf} \leq \text{mid} \leq \text{sup}, \text{true}$	✓
$\alpha[\text{mid}] < x$	non sta in $\alpha[\text{inf} \dots \text{mid}]$	
quindi: $x \in \alpha[\text{inf} \dots \text{sup}]$		

↑
 $x \in [\text{mid}+1 \dots \text{sup}]$

Per l'ip. induttiva

binary-search ($x, \alpha, \text{mid}+1, \text{sup}$) è corretta, cioè (sse)
ovvero da t/f in base ?.

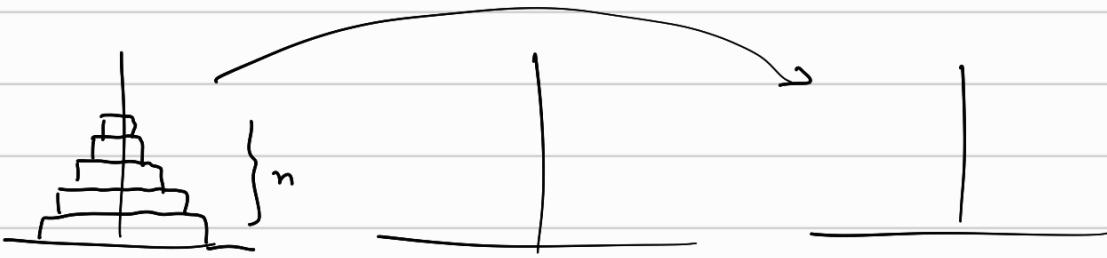


binary-search ($x, \alpha, \text{inf}, \text{sup}$) vale per l'alg completo

Analogamente x il 2° caso

Esempio delle Torri di Hanoi

- mostra la "potenza della ricorsione"
- esempio di problema intrattabile (solo esponenziale)



Caso base: Sappiamo risolvere il problema per $n=1$? sì

Assumiamo di saper spostare $n-1$ dischi (hp. ind.) (la ricorsione parte dal basso (casusimo e la vette))
 → sposta $n-1$ nel piolo intermedio
 → sposta l'ennesimo nel terzo e poi prolli nel secondo nel terzo

Hanoi (n , from, to, aux)

if ($n=1$) move (from, to) // non importa i passi che esegue questa funzione
 else

Hanoi ($n-1$, from, aux)
 move (from, to)
 Hanoi ($n-1$, aux, to)

Complessità (contiene proprio il n. di mosse $T(n)$ per n dischi)

$$T(1) = 1$$

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

$$n \geq 1$$

|| RECURSIONE
DI
RICORRENZA

(complessità uguale
in "modo riversivo")

Metodo "per sostituzioni successive"

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

$$= z \left[z T(n-2) + 1 \right] + 1 = 1 + z + z^2 T(n-2)$$

$$= 1 + z + z^2 \left[z T(n-3) + 1 \right] = z^0 + z^1 + z^2 + z^3 T(n-3)$$

$$= \dots = z^0 + z^1 + z^2 + \dots + z^i T(n-i)$$

mi forma per $i = n-1$ ($\rightarrow T(1) = 1$)

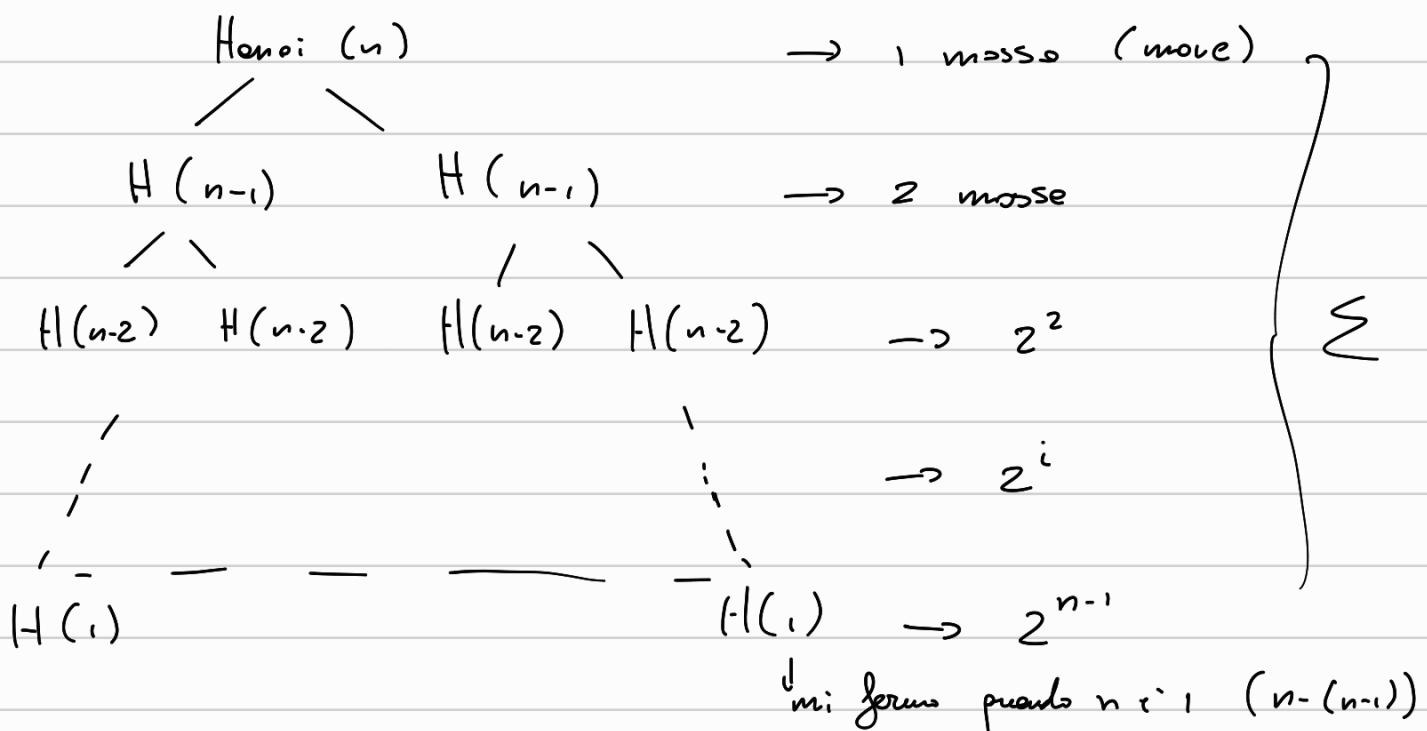
$$= z^0 + z^1 + z^2 + \dots + z^{n-1}$$

$$= \sum_{i=0}^{n-1} 2^i = \frac{2^{n-1}}{2-1} = \underbrace{2^{n-1}}_{\text{esponente}}$$

serie geometrica

$$\sum_{i=0}^{n-1} q^i = \frac{q^n - 1}{q - 1}$$

Altro modo di ragionare (ALBERO DI RICORSIONE)



Prova (di controllo) rigorosa per induzione aritmetica

Tesi $T(n) = 2^n - 1 \quad \forall n \geq 0$

Base $T(0) = 2^0 - 1 = 1$ ok

$$\begin{aligned} T(n+1) &= \underset{\text{def}}{2T(n)+1} = \underset{\text{H.P.I.}}{2 \cdot (2^n - 1) + 1} \\ &= 2^{n+1} - 2 + 1 = 2^{n+1} - 1 \end{aligned}$$

Abbiamo dato un semplice alg ricorsivo con correttezza $\in \Theta(2^n)$
Si puo' provare (sempre seguendo per induzione) che non possiamo mettere
 $< 2^n - 1$ mosse

$(n=1)$ 1 mossa (e non di meno)



dovrò prima spostare gli $n-1$ in aux
per hp ind ci vogliono almeno $2^{n-1} - 1$ mosse
(posso mettere cose di più)

Dunque c'è un problema inaffidabile.

(vi è un'alternativa iterativa, ma le mosse sono le medesime del
ricorsivo)

Esempio di soluzione di relazione di ricchezza

Ricerca binaria

$$T(1) = 1 \quad (\exists(1))$$

$$T(n) = 1 + T(n/2)$$

Conviene porre
 $n = 2^k$

$$T(2^0) = 1$$

$$\begin{aligned} T(2^k) &= 1 + \overline{T(2^{k-1})} \\ &= 1 + 1 + T(2^{k-2}) \end{aligned}$$

= ...

$$= i + T(2^{k-i})$$

mi fermo quando $k-i=0$
 $\rightarrow i=k$

$$= k+1 = \log n + 1$$

quicksort(s)

if $n > 1$

estraggo p da s (es. il primo)

$s_1 \leftarrow$ elementi $\leq p$

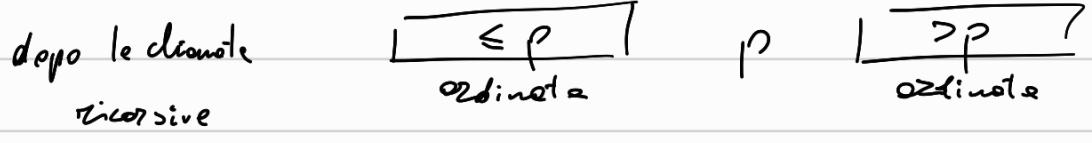
$s_2 \leftarrow$ " $> p$

quicksort(s_1) • p • quicksort(s_2) concatenazione
return $s_1 \cdot p \cdot s_2$ con s_1 e s_2 ordinate

Correttezza per induzione

Base $n \leq 1$ (vuota o sol.) l'elg non fa nulla OK

$n > 1$ $\boxed{\leq p} / \quad p \quad \boxed{> p}$



$|s_1| < |s|$ per cui abbiamo tolto (se avessi $>s$ l'algoritmo non
 $|s_2| < |s|$ il pivot termina)

Keywords:

Torre di Hanoi

Relazione di ricorrenza

Sostituzione successiva

Serie geometrica

Complessità:

$$T(1) = 1$$

$$T(n) = n + T(r) + \overbrace{T(n-r-1)}^{\substack{\text{scelta pivot} \\ \text{due sottosequenze}}} \quad r \geq 0 \quad \text{diverso A} \quad \text{chiama} \\ \text{(divisione diversa} \\ \text{ogni volta)}$$

CASO PEGGIORI

Ogni divisione \rightarrow sottosequenza sbilanciata da una parte

$$\begin{aligned} T(n) &= n + T(n-1) + \cancel{T(0)} \\ &= n + n-1 + \cancel{T(n-2)} = \dots \\ &= n + (n-1) + (n-2) + \dots + 1 \\ &= \frac{n(n+1)}{2} = \mathcal{O}(n^2) \end{aligned}$$

CASO MIGLIORE

soluzione = metà sequenza

$$T(n) = n + 2T\left(\frac{n}{2}\right) \quad (\text{come mergesort})$$

Assumo $n = 2^k$

$$T(2^0) = 1$$

$$\begin{aligned} T(2^k) &= 2^k + 2T(2^{k-1}) \\ &= 2^k + 2[2^{k-1} + 2T(2^{k-2})] \\ &= 2^k + 2^k + 2^2 T(2^{k-2}) \\ &= \underbrace{2^k + \dots + 2^k}_{i \text{ volte}} + 2^i T(2^{k-i}) \quad \text{m: forme con } i=k \\ &\qquad \qquad \qquad T(2^{k-i}) = 1 \end{aligned}$$

$$= \underbrace{2^k + \dots + 2^k}_{k+1 \text{ volte}} = (k+1)2^k = \mathcal{O}(n \log n)$$

$$\begin{aligned} \log_2(2^k) &= k \\ n &\qquad \qquad \qquad (\log n + 1)n \\ &\qquad \qquad \qquad \underbrace{n \log n}_{\text{dominante}} + n \end{aligned}$$

Esercizio - controllo a involv. aritmetica

CASO PEGGIORI

$$T(\text{caso peggiore}) : T(n) = \frac{n(n+1)}{2} = \mathcal{O}(n^2) \quad \text{con } n \geq 0$$

$$\text{Base } T(1) = 1$$

$$\begin{aligned} T(n+1) &= n+1 \quad T(n+1) + T(0) \\ &\stackrel{\text{def}}{=} n+1 \quad n+1 + T(n+1-1) \\ &= 2n+2 + n+1 + T(n-1) = 3n+2 + n-1 + T(n-2) \\ &= 4n+1 \dots - 1 \end{aligned}$$

$$\text{owers } 3(n) + 2 + (n-1) + (n-2) \dots 1$$

{ele sommatoria role} $\frac{(n+1)(n+1-1)}{2} = \frac{n(n+1)}{2} = \mathcal{O}(n^2)$ ✓

Also negl. $T(n) = 2^{\alpha}(k+1) = \mathcal{O}(n \log n)$

Base $T(1) = 1$

$$T(n) \underset{\text{def}}{=} n + 2T\left(\frac{n}{2}\right) \rightarrow T(n+1) = n+1 + 2T\left(\frac{n+1}{2}\right)$$

ossunto $n+1 = 2^k \rightarrow T(2^k) = 2^k + 2T(2^{k-1})$
 $= 2^k + 2[2^{k-1} + 2T(2^{k-2})]$
 $= 2^k + 2^k + 2^2 T(2^{k-2})$
 $= 2^k + 2^k + 2^k + 2^3 T(2^{k-3}) = \dots$

$\underbrace{\quad}_{i \text{ volte}} \quad \text{mi fa una per } i=k$

$T(2^{k-i}) = 1$

$= 2^k + \dots + 2^k + 2^i T(2^{k-i})$
 $= 2^k \underbrace{\dots + 2^k}_{i \text{ volte}} + 2^i T(2^{k-i})$
 $= 2^k (k+1) = \mathcal{O}((n+1) \log(n+1))$

$$\mathcal{O}((n+1) \log(n+1))$$

$$= \mathcal{O}(n \log n) \quad \checkmark$$