Una lezione (quasi) universitaria di Informatica.
Liceo Scientifico Statale "Galileo Ferraris".

Torino, 27 aprile 2007

Parte IV

Elio Giovannetti
Dipartimento di Informatica
Università di Torino

Università di Torino – Facoltà di Scienze MFN
Corso di Studi in Informatica
Curriculum SR (Sistemi e Reti)

Algoritmi e Laboratorio a.a. 2006-07
Lezioni

prof. Elio Giovannetti

Parte 13 – Le Torri di Hanoi
versione 28/11/2006
adattata al Pascal per lezione al Galfer

- Il gioco delle torri di Hanoi è costituito da tre pioli verticali (sinistro, centrale, destro), e da un certo numero di dischi forati, di diametri tutti diversi fra di loro, che si possono infilare nei pioli.
- All'inizio del gioco tutti i dischi sono infilati su un piolo, ad esempio il sinistro, in ordine decrescente di diametro dalla base verso la cima del piolo.
- Lo scopo del gioco è quello di spostare tutti i dischi su un altro piolo (ad es. sul destro), eventualmente utilizzando il piolo rimanente (ad es. il centrale), senza mai violare la seguente regola:
  - si può spostare da un piolo ad un altro un solo disco per volta, cioè quello più in alto nel piolo di partenza;
  - non si può posare un disco sopra un altro disco di diametro inferiore.

Problema

Il gioco è risolubile per qualunque numero n di dischi ?

Se lo è, qual è per ogni n la sequenza minima di mosse che lo risolve ?

Premessa (ovvia)

Lo spostamento di un disco singolo da un piolo ad un altro, se è possibile secondo le regole del gioco (cioè se non ha dischi sopra di sé, e se viene spostato sopra un disco più grande di sé), non richiede l'uso di un piolo ausiliario.

Teorema:

Il gioco è risolubile per qualsiasi numero n di dischi.

La dimostrazione è per induzione su n.

Dimostrazione.

Base dell'induzione: Il gioco è risolubile per n = 1.

Dimostrazione: Ovvia, perché una torre di altezza 1, cioè un solo disco, può essere spostata con una sola mossa da un piolo ad un altro (senza bisogno di piolo ausiliario).

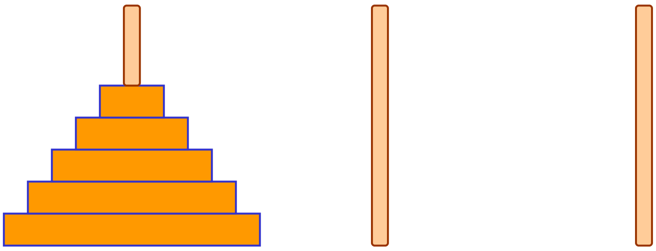
Passo di Induzione:

Ipotesi (induttiva):
Il problema di spostare una torre di altezza k-1 da un qualunque piolo ad un qualunque altro piolo con l'ausilio di un piolo ausiliario è risolubile.

Tesi:
Il problema di spostare una torre di altezza k da un piolo (qualunque) orig ad un altro piolo dest con l'ausilio di un terzo piolo aus è risolubile.

Dimostrazione (continua).

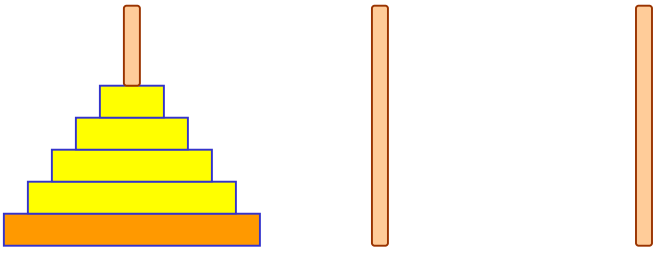
Dimostrazione del Passo:



Dimostrazione (continua).

Dimostrazione del Passo:

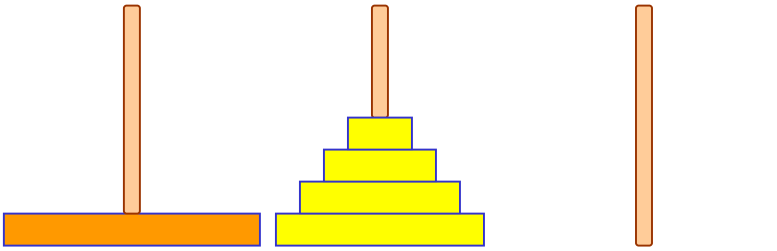
• si spostano i k-1 dischi superiori



Dimostrazione (continua).

Dimostrazione del Passo:

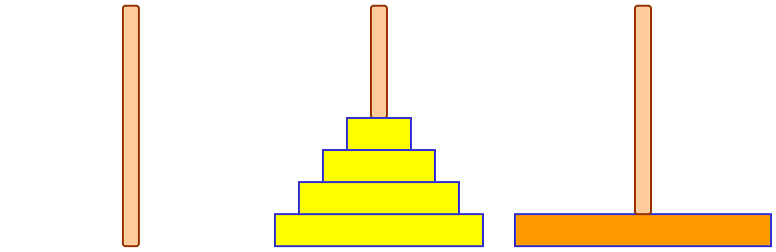
- si spostano i k-1 dischi superiori dal piolo orig al piolo aus con l'ausilio di dest (è possibile per ipotesi induttiva);



Dimostrazione (continua).

Dimostrazione del Passo:

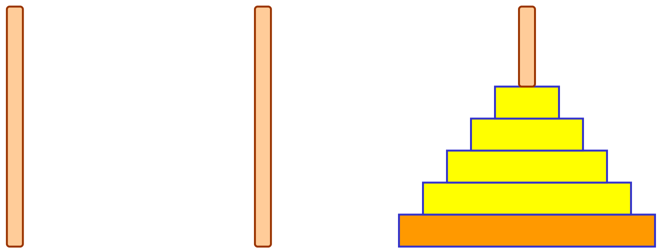
- si sposta il restante disco grande da orig a dest;



Dimostrazione (continua).

Dimostrazione del Passo:

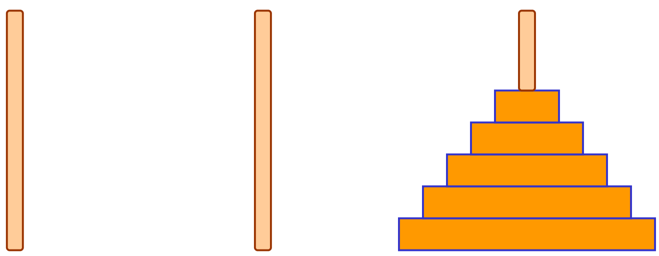
- si spostano i k-1 dischi dal piolo aus al piolo dest con l'ausilio di orig (è possibile per ipotesi induttiva).



Dimostrazione (continua).

Dimostrazione del Passo. Riassumendo:

- si spostano i k-1 dischi superiori dal piolo orig al piolo aus con l'ausilio di dest (è possibile per ipotesi induttiva);
- si sposta il restante disco grande da orig a dest;
- si spostano i k-1 dischi dal piolo aus al piolo dest con l'ausilio di orig (è possibile per ipotesi induttiva).



La dimostrazione induttiva della risolubilità del problema fornisce direttamente il metodo ricorsivo di risoluzione.

È quindi immediato scrivere la corrispondente procedura.

La procedura ricorsiva.

```
type piolo = (sinistro, centrale, destro);
type posint = 1 .. maxint;
  ...
procedure hanoi(n: posint; p1, p2, p3: piolo);
  begin
    if n = 1 then sposta disco da p1 a p3
    else begin
      hanoi(n-1, p1, p3, p2);
      sposta disco da p1 a p3;
      hanoi(n-1, p2, p1, p3)
    end
  end;
```

La procedura ricorsiva (schema di base)

```
type piolo = (sinistro, centrale, destro);
type posint = 1 .. maxint;
  ...
  procedure hanoi(n: posint; p1, p2, p3: piolo);
  begin
    if n = 1 then spostadisco(p1, p3)
    else begin
      hanoi(n-1, p1, p3, p2);
      spostadisco(p1, p3);
      hanoi(n-1, p2, p1, p3)
    end
  end;

begin
  write('Quanti dischi ? '); readln(ndischi);
  hanoi(ndischi, sinistro, centrale, destro); ...
end.
```


http://www.cut-the-knot.org/recurrence/hanoi.shtml

Analisi della complessità: equazioni di ricorrenza

Il tempo di calcolo è evidentemente proporzionale al numero delle istruzioni di spostamento di disco, cioè al numero di mosse necessarie per risolvere il gioco.

Per risolvere il gioco per n = 1 occorre una mossa: $T(1) = 1$
Per risolvere il gioco per n > 1 occorrono:

- $T(n-1)$ mosse per spostare gli n-1 dischi da orig a aus;
- 1 mossa per spostare il disco da orig a dest;
- $T(n-1)$ mosse per ri-spostare gli n-1 dischi da aus a dest;

Le equazioni di ricorrenza sono allora:

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

Risoluzione delle equazioni di ricorrenza

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

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

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

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

$= 2^2T(n-2) + 2 + 1 =$

$$T(n-2) = 2T((n-2)-1) + 1 = 2T(n-3) + 1$$

$= 2^2(2T(n-3) + 1) + 2 + 1 =$

$= 2^3T(n-3) + 2^2 + 2 + 1 =$

...

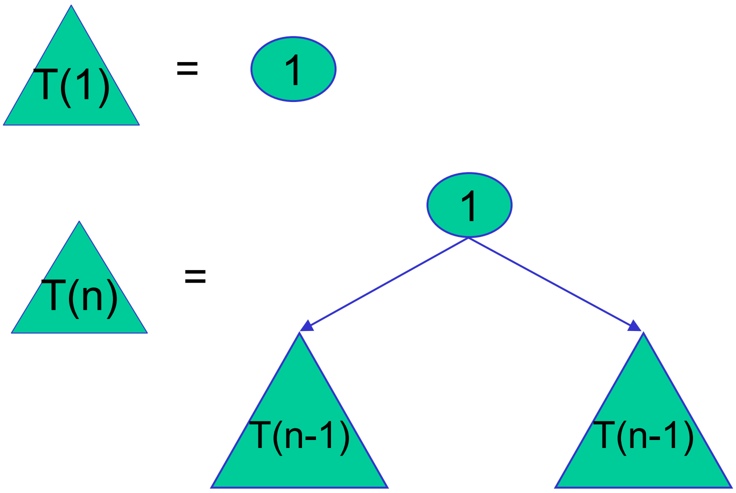
$= 2^kT(n-k) + 2^{k-1} + 2 + 1 =$

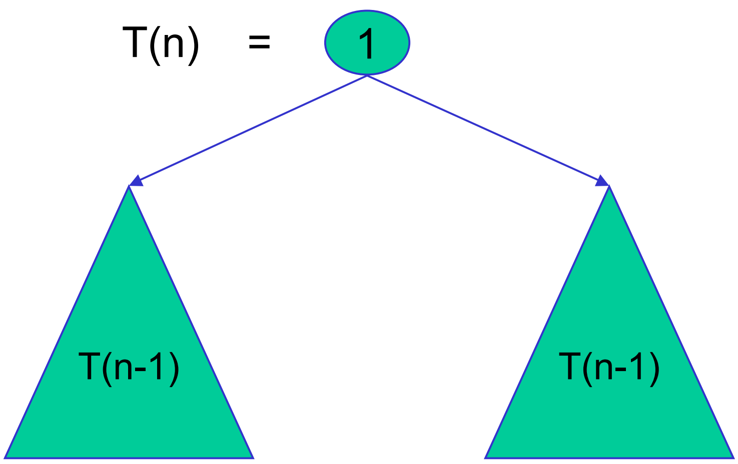
per $n-k = 1$, cioè $k = n-1$:

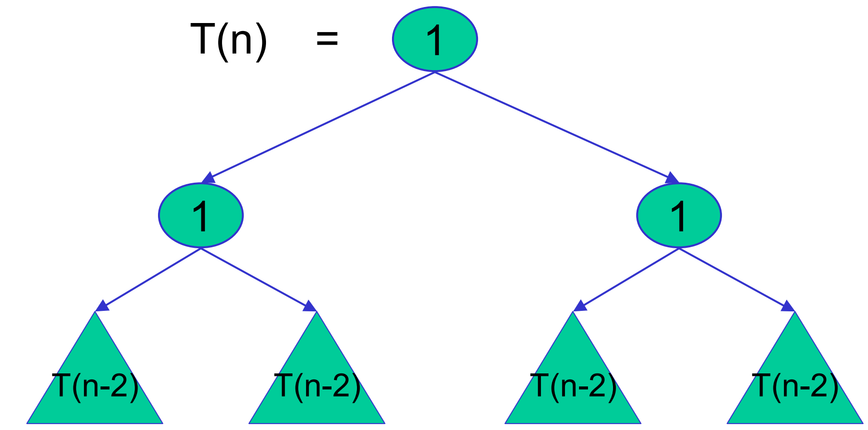
$T(n) = 2^{n-1}T(1) + \ldots + 2^2 + 2 + 1 = 2^{n-1} + \ldots + 2^2 + 2 + 1 = 2^n - 1$

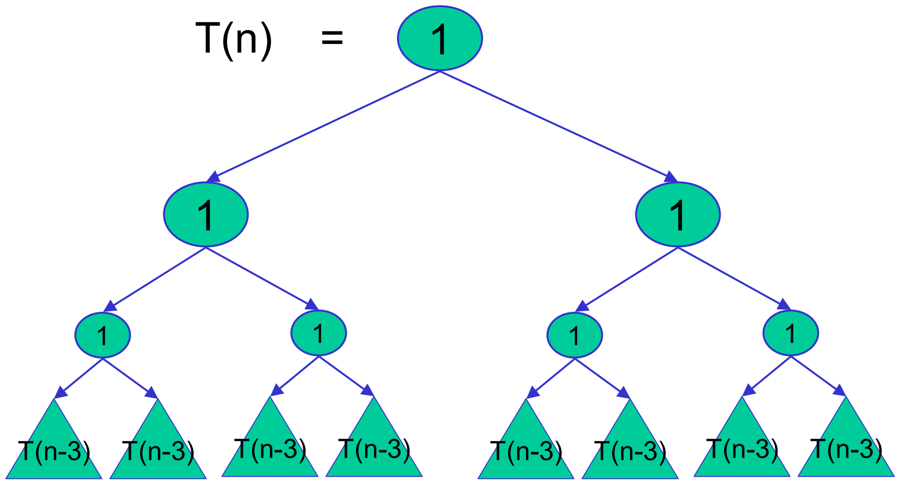
$$T(n) = 2^n - 1$$

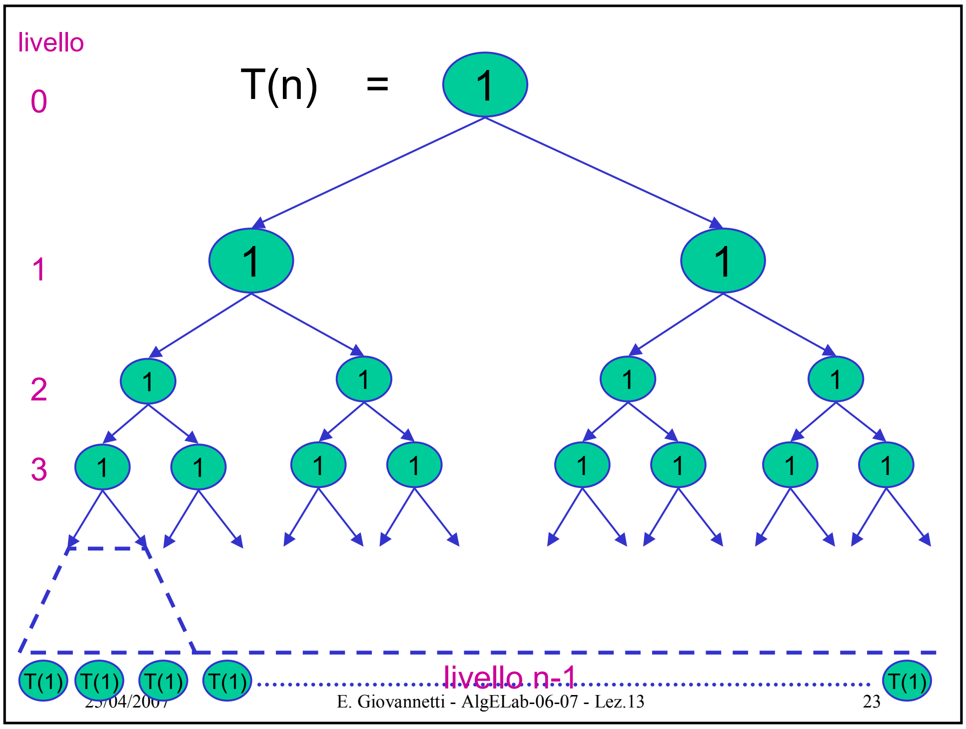
Rappresentazione grafica delle eq. di ricorrenza (albero di ricorsione)

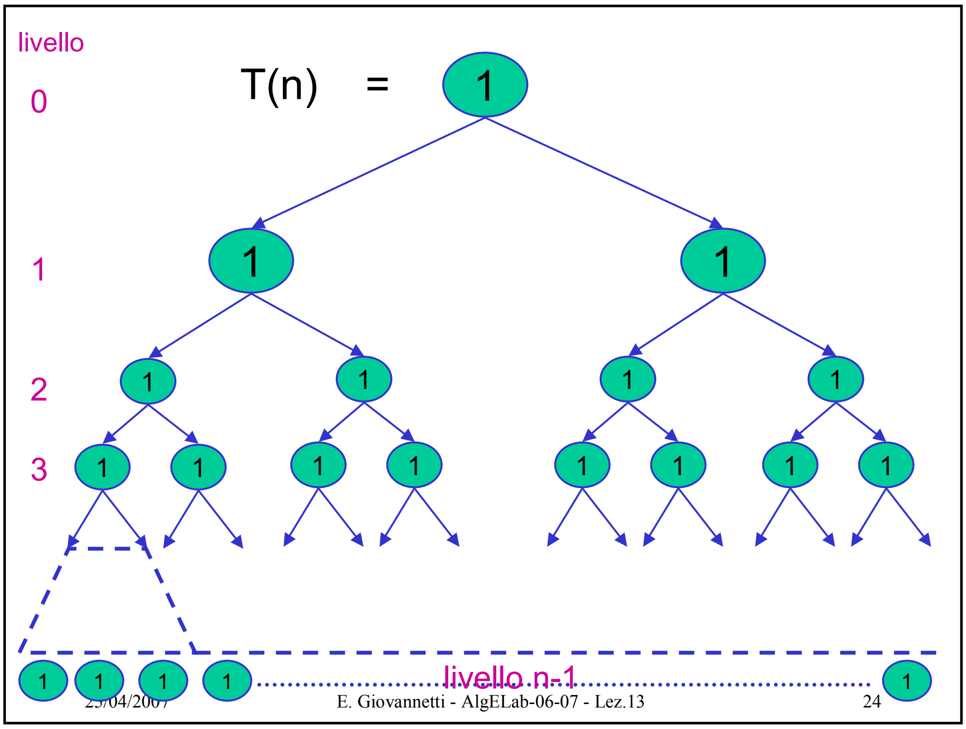





T(n)
=
1
1
1
T(n-2)
T(n-2)
T(n-2)
T(n-2)


T(n)
=
1
1
1
1
1
1
1
T(n-3)
T(n-3)
T(n-3)
T(n-3)
T(n-3)
T(n-3)
T(n-3)
T(n-3)


livello
0
1
2
3
T(n) =
1
1 1
1 1 1 1
1 1 1 1 1 1 1 1
T(1) T(1) T(1) T(1) T(1)
livello n-1
E. Giovannetti - AlgELab-06-07 - Lez.13
23


livello
0
1
2
3
T(n) =
1
1 1
1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1
livello n-1
E. Giovannetti - AlgELab-06-07 - Lez.13
24

Ogni nodo vale 1, quindi il tempo totale è il numero dei nodi dell'albero di ricorsione.
L'albero è un albero completo a n livelli, da 0 a n-1;
il numero dei nodi è quindi $2^n - 1$.

La complessità in spazio è data dall'altezza dell'albero di ricorsione, che rappresenta la massima profondità dello stack.
Abbiamo quindi:

$$T(n) = \Theta(2^n) \qquad S(n) = \Theta(n)$$

Il gioco delle Torri di Hanoi è intrattabile, poiché il numero di mosse costituenti la soluzione cresce esponenzialmente al crescere del numero dei dischi. È del tutto ovvio che l'algoritmo descritto è l'unico possibile (eccetto banali varianti, quali quella ottenuta eliminando la ricorsione di coda, che vedremo).