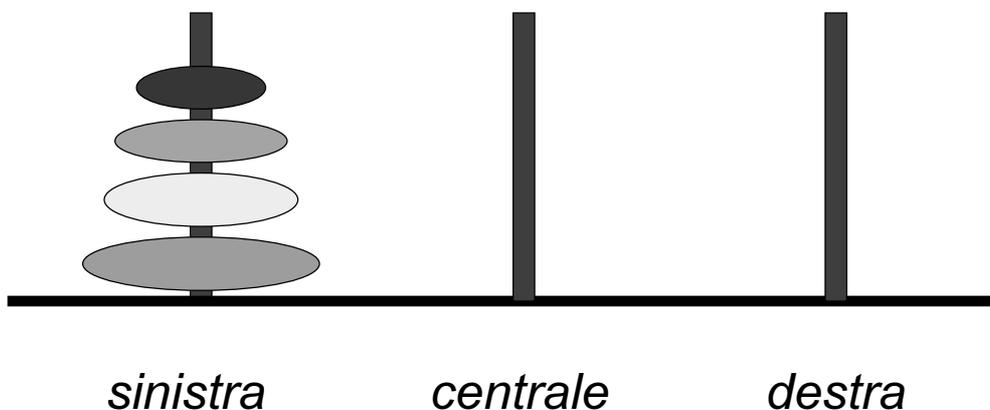


## ESERCIZIO: la Torre di Hanoi

- Sono date tre torri (*sinistra*, *centrale*, e *destra*) e un certo numero  $N$  di dischi forati.
- I dischi hanno diametro diverso gli uni dagli altri, e inizialmente sono infilati uno sull'altro (dal basso in alto) dal più grande al più piccolo sulla torre di sinistra.
- Scopo del gioco è *portarli tutti sulla torre di destra*, rispettando due regole:
  - a) si può muovere un solo disco alla volta;
  - b) un disco grande non può mai stare sopra un disco più piccolo.

## ESERCIZIO: la Torre di Hanoi



## ESERCIZIO: la Torre di Hanoi

### Come risolvere il problema?

- Immaginare la serie di mosse che, *in generale*, risolve il problema è impensabile
- Invece, è abbastanza semplice esprimere una soluzione ricorsiva

### Ipotesi:

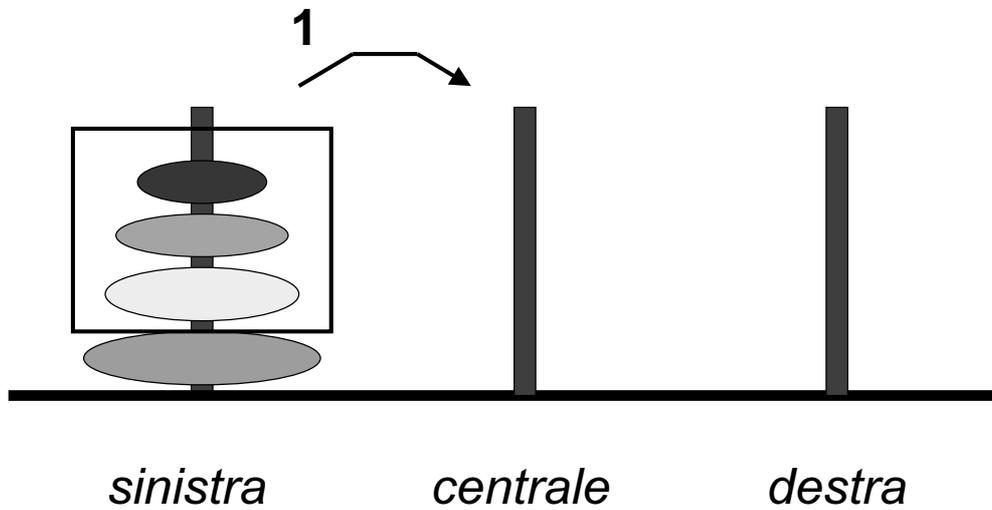
- Siamo capaci di spostare un singolo disco tra due torri a scelta.

## ESERCIZIO: la Torre di Hanoi

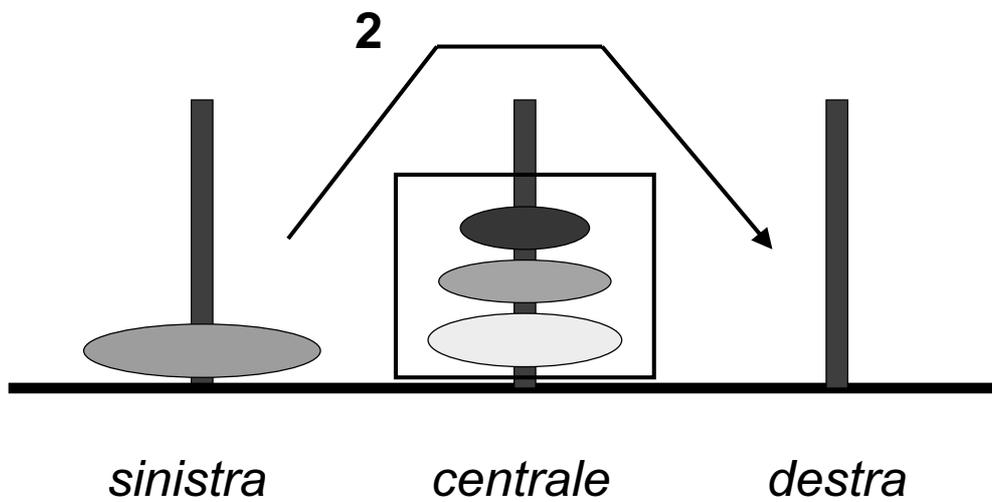
### Soluzione ricorsiva

- Caso banale: un singolo disco si sposta direttamente dalla torre iniziale a quella finale
- Caso generale: per spostare  $N$  dischi dalla torre iniziale a quella finale occorre
  - spostare  $N-1$  dischi dalla torre iniziale a quella intermedia, che funge da appoggio
  - spostare il disco rimanente (il più grande) direttamente dalla torre iniziale a quella finale
  - spostare gli  $N-1$  dischi “posteggiati” sulla torre intermedia dalla torre intermedia a quella finale

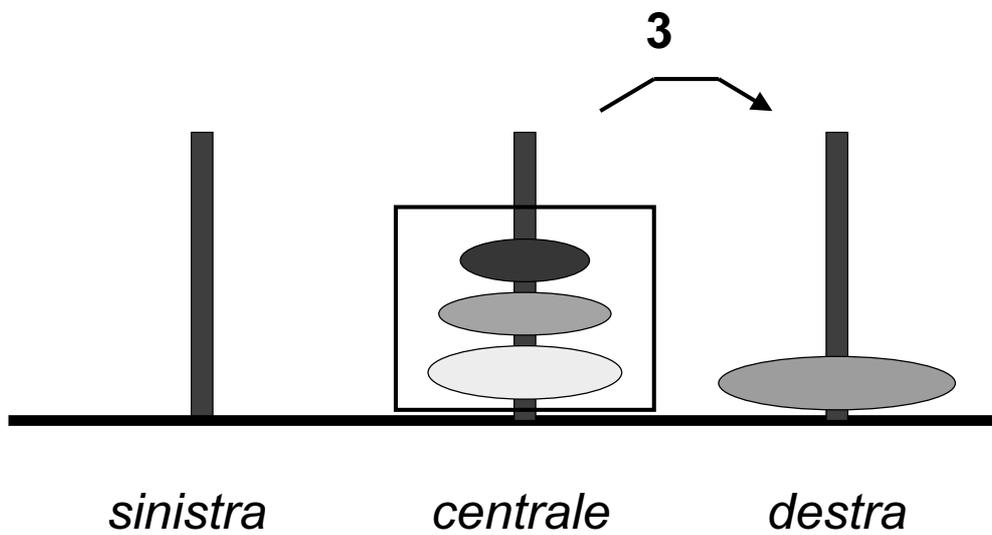
## ESERCIZIO: la Torre di Hanoi



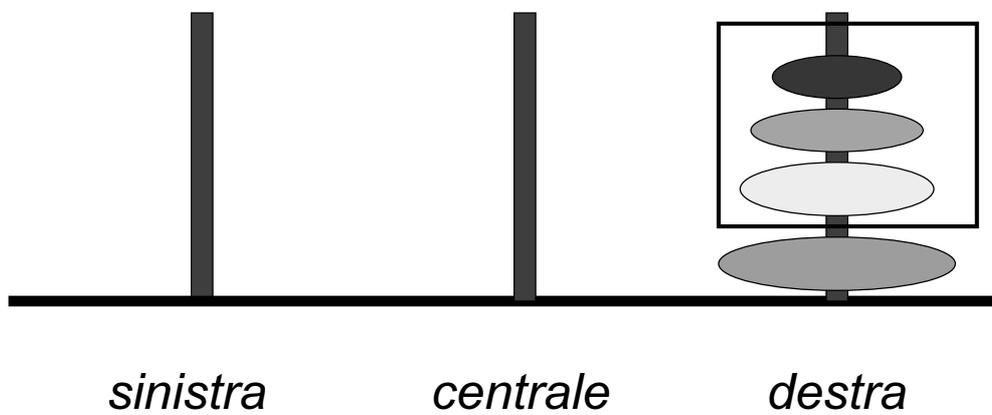
## ESERCIZIO: la Torre di Hanoi



## ESERCIZIO: la Torre di Hanoi



## ESERCIZIO: la Torre di Hanoi



## ESERCIZIO: la Torre di Hanoi

La soluzione delineata per il caso “N dischi” presuppone

- di sapere spostare N-1 dischi  
→ *stesso problema in un caso più semplice*
- di sapere spostare un singolo disco  
→ *abilità che possediamo per ipotesi.*

È una ricorsione non lineare

- il problema con N dischi si espone in due sottoproblemi con N-1 dischi
- con N dischi,  $2^N-1$  attivazioni della funzione.

## ESERCIZIO: la Torre di Hanoi

### Specifica

- rappresentiamo le tre torri con un intero
- rappresentiamo ogni mossa tramite *la coppia di torri coinvolte* (in futuro le scriveremo sull'output)
- la funzione `hanoi` ha come parametri
  - *il numero di dischi da spostare*
  - *la torre iniziale*
  - *la torre finale*
  - *la torre da usare come appoggio*
- non ha tipo di ritorno, è una procedura → `void`

## ESERCIZIO: la Torre di Hanoi

### Interfaccia

```
void hanoi(int dischi, int torreIniziale,  
           int torreFinale, int torreTransito);
```

### Codifica

```
void hanoi(int dischi, int iniziale,  
           int finale, int transito){  
    if (dischi==1) {  
        /* muovi da iniziale a finale */  
    } else {  
        hanoi(dischi-1, iniziale, transito, finale);  
        /* muovi da iniziale a finale */  
        hanoi(dischi-1, transito, finale, iniziale);  
    }  
}
```

## ESERCIZIO: la Torre di Hanoi

### Alcune domande

- come avremmo fatto senza la ricorsione?
- senza debugger, cosa "computa" alla fine?

### Alcune proposte

- costruire un idoneo cliente e provare il programma
- provare, in un caso semplice, a seguire quello che accade passo per passo
- in futuro, inserire al posto dei commenti delle istruzioni di stampa a video.

