Il problema della Torre di Hanoi

Ivano Salvo
Sapienza Università di Roma
email: salvo@di.uniroma1.it

21 novembre 2012

Forse vi siete già imbattuti nel problema della *Torre di Hanoi*:

> narra la leggenda che in un tempio Indù alcuni monaci siano costantemente impegnati a spostare su tre colonne di diamante 64 dischi d'oro di diversi diamentri secondo le regole della Torre di Hanoi (a volte chiamata Torre di Brahma): ogni disco può essere spostato da una colonna all'altra senza mai che un disco di diametro maggiore sia posto sopra un disco di diametro inferiore. I monaci spostano un disco ogni giorno e quando completeranno il lavoro, il mondo finirà.

In Fig. 1 è raffigurato lo stato iniziale del problema. Il rompicapo può banalmente essere risolto con un solo disco. Due facili mosse lo risolvono nel caso di due dischi. Anche andando a tentativi, probabilmente riuscirete a risolverlo anche nel caso di 3 dischi. Tuttavia, al crescere della dimensione del problema, è necessaria una strategia ben precisa. Confortati dalla soluzione dei casi facili, si può elegantemente osservare che avendo da spostare k dischi dal piolo 1 (o *sorgente*) al piolo 3 (o *destinazione*), usando il piolo 2 come *appoggio*, sarà sufficiente spostare $k-1$ dischi dal piolo sorgente al piolo ausiliario, spostare il disco di diametro maggiore rimasto finalmente libero di muoversi dal disco sorgente a quello destinazione, ed infine spostare i $k-1$ dischi dal piolo appoggio al piolo destinazione (sopra il disco di diametro maggiore) usando il piolo sorgente come appoggio (vedi Fig. 2).

Abbiamo risolto il problema della Torre di Hanoi per un qualsiasi numero di dischi, semplicemente riconducendo una generica istanza del problema con k dischi alla soluzione di due istanze dello stesso problema con $k-1$ dischi e una istanza banale con 1 disco solo da muovere. L'unica verifica da fare è che lo spostamento di $k-1$ dischi non viola le regole del gioco, ma ciò è vero, perché l'unico altro disco rimasto è quello di diametro maggiore e quindi può essere ignorato: ogni altro disco gli può essere appoggiato sopra. L'ultima osservazione da fare, prima di scrivere un programma che risolve questo problema, è che i ruoli dei pioli 1, 2 e 3 cambiano durante la soluzione: ad esempio, il piolo 1 che è il piolo sorgente, viene usato come

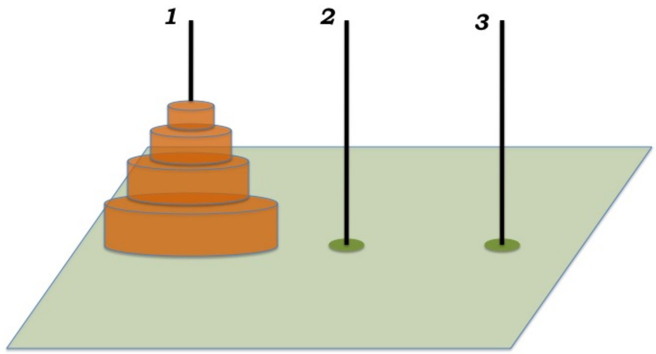

Figura 1: Il problema della Torre di Hanoi con 4 dischi

piolo di appoggio nella soluzione della seconda istanza con $k-1$ dischi. Occorre quindi parametrizzare la nostra soluzione, dimodochè sappia spostare in generale k dischi da un certo piolo sorgente a un piolo destinazione usando un terzo piolo come ausiliario, indipendentemente da quali siano effettivamente i numeri che identificano il piolo sorgente, destinazione e appoggio.

```
void hanoi(int sorg, int aux, int dest, int n){
/* sposta n dischi dal piolo sorg al piolo dest, usando aux come appoggio
 */
     if (n==1) muovi(sorg, dest);
        else {
            hanoi(sorg, dest, aux, n-1);
            muovi(sorg, dest);
            hanoi(aux, sorg, dest, n-1);
            }
     }
```

Abbiamo mantenenuto una certa flessibilità su cosa fare una volta scoperta la mossa giusta da fare, invocando la funzione `void muovi(int s, int d)` con parametri `sorg` e `dest`. Potrebbe essere l'animazione a video del disco che abbandona il piolo `s` e si muove verso il piolo `d`, oppure la stampa testuale della situazione dei tre pioli (lo vedremo nel prossimo futuro quando avremo abbastanza strutture dati per rappresentare lo stato del gioco – cosa che per fortuna abbiamo visto non essere necessaria) oppure potremo limitarci a stapare la mossa:

```
void muovi(int s, int d){
   printf("%d --> %d\n",s,d);
   }
```

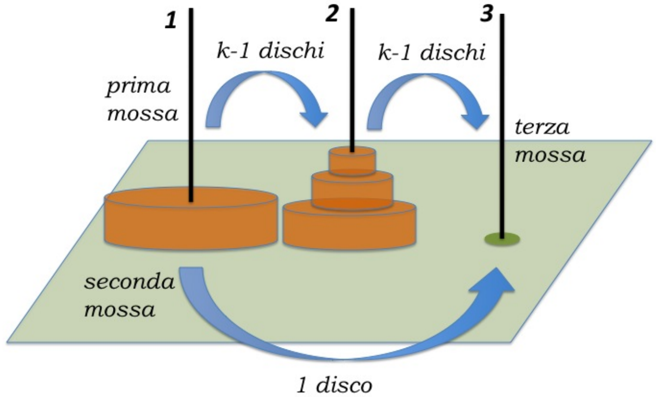

Figura 2: Soluzione del problema della Torre di Hanoi

Di questo algoritmo, non esiste una semplice versione iterativa.