

# Esercitazione N°1

Luca Zepponi

26 ottobre 2022

## 1. DESCRIZIONE DEL PROGRAMMA

---

Il programma prende in input il numero di dischi con cui si vuole giocare al rompicapo della Torre di Hanoi, lo risolve fornendo la rappresentazione della torre a inizio partita e il tempo di esecuzione.

La strategia utilizzata è quella ricorsiva.

Le librerie utilizzate sono:

- `<iostream>`: per gli input e gli output;
- `<time.h>`: per misurare il tempo di esecuzione;
- `<string>`: per poter utilizzare l'omonima classe.

I vari output sono stati messi in modo da obbligare il testo ad andare a capo prima di svilupparsi troppo in lunghezza.

### 1.1. DESCRIZIONE DELLE SINGOLE FUNZIONI

---

Il programma riportato nella sezione 2 inizia con un piccolo testo dove viene fatta una rapida presentazione del gioco, della sua storia, dell'obiettivo e comprende anche le regole da sapere per poter giocare.

Dopo questo piccolo incipit viene subito richiesto all'utente di inserire da tastiera il numero di dischi con cui desidera giocare e lo immagazzina nella variabile `nDischi`. Tale numero è soggetto a delle restrizioni:

- infatti qualora si immettesse un numero negativo (o nullo), il gioco non potrebbe partire per ovvie ragioni;
- anche nel caso in cui si desiderasse giocare con più di 10 dischi, il programma obbligherebbe l'utente a scegliere un numero più basso per motivi di risorse.

Immettere un numero che soddisfi contemporaneamente queste due condizioni (ovvero il numero di dischi deve essere compreso fra 1 e 10) è essenziale per il proseguimento, altrimenti il programma continua a richiedere un numero con le suddette caratteristiche.

Una volta dato un numero accettabile di dischi, prima di passare alla vera risoluzione del gioco, viene fornito in output una schematizzazione della situazione iniziale. Per disegnare correttamente la torre di Hanoi con il giusto numero di dischi ho inizializzato due stringhe, rispettivamente **disco** e **base** che serviranno per disegnare rispettivamente i dischi e la base.

Per quanto riguarda i dischi, ho deciso di creare un ciclo **for** che si ripete tante volte quanto è l'interno in **nDischi**. All'interno del ciclo sono annidati altri due **for**, il primo che aggiunge il numero corretto di - sia a sinistra che a destra, mentre il secondo inserisce la giusta spaziatura in modo da centrare il centro del disco con la "rappresentazione" del primo piolo. A questo fine, l'indice **j** si deve fermare a **1+nDischi-k** (e non a **nDischi-k**) perché sulla base è stata inserita una barra verticale per indicare l'inizio.

Per rappresentare la base ho seguito un ragionamento simile. Il **for** disegna solo il primo terzo della struttura aggiungendo tanti trattini orizzontali alla destra e alla sinistra del piolo quanti sono i dischi più uno. Per aggiungere gli altri due terzi della base ho semplicemente concatenato la base tre volte frapponendo la stringa **base** con dei trattini verticali in modo da evidenziare la separazione fra le tre parti.

Arrivati a questo punto inizia la risoluzione del gioco. Tale parte si basa su una funzione ricorsiva che prende in input il numero di dischi inserito all'inizio e i tre pioli del gioco, dove:

- il primo piolo, ovvero quello di partenza, è stato indicato con la stringa "**a sinistra**";
- il piolo di arrivo è stato indicato con la stringa "**a destra**";
- il piolo centrale, nel codice chiamato anche come ausilio, è stato indicato con la stringa "**al centro**".

All'interno della funzione **Hanoi** è stato inserito un contatore, inizializzato globalmente a 1, per contare il numero di mosse che l'algoritmo impiega e viene richiamata la funzione **spostamento** che si occupa di mandare a video le istruzioni da seguire per giungere alla soluzione. Quest'ultima funzione prende in input le stesse variabili della funzione **hanoi**.

Il contatore incrementa il suo valore dopo che viene mandato in stampa.

La funzione **hanoi** sposta i primi  $n - 1$  dischi da **partenza** ad **ausilio**, poi sposta il disco **n** da **partenza** a **arrivo** e infine sposta  $n - 1$  dischi da

ausilio a arrivo. Tuttavia, la funzione non può spostare  $n$  dischi (perché violerebbe le regole del gioco), quindi occorre fare un'operazione elementare, ovvero spostare un singolo disco ed una complessa, ossia spostare  $n - 1$  dischi. Tale operazione ha la stessa strategia risolutiva: si richiede come operazione complessa lo spostamento di  $n - 2$  dischi. Iterando questo algoritmo si arriva dal processo complesso al processo elementare.

Una volta che la funzione `Hanoi` ha spostato tutti i dischi, uno alla volta, rispettando le condizioni imposte, l'algoritmo stampa il tempo di esecuzione che ci impiega il programma. In tale tempo non viene però considerato l'intervallo temporale che l'utente impiega per inserire il numero di dischi e il relativo controllo.

Per misurare il tempo di esecuzione del programma è stata utilizzata la libreria `<time.h>` per poter inserire le variabili `clock_t start = clock();` e `clock_t end = clock();` per far partire e stoppare il cronometro, mentre per mandare a schermo il tempo impiegato è stato sufficiente inserire un `printf`. Dato l'esiguo tempo di esecuzione è stato effettuato un casting per aumentare la precisione del cronometro. Nella libreria `<time.h>`:

- vi è definito il tipo `clock_t` per la memorizzazione dei cicli di clock;
- è anche presente la costante `CLOCK_PER_SEC` che restituisce il numero di clock per secondo per la piattaforma hardware e software che si sta usando.

## 2. CODICE SORGENTE

---

```
1  #include <iostream>
2  #include <time.h>
3  #include <string>
4
5  // Funzioni che verranno utilizzate in seguito
6  void hanoi(int, std::string, std::string, std::string);
7  void spostamento(int, std::string, std::string, std::string);
8
9  int counter = 1; // Contatore per la funzione Hanoi.
10 // Conta il numero di mosse che la funzione esegue
11
12 using namespace std;
13
14 int main(){
```

```

15     setlocale (LC_ALL, "");
16
17     // Dichiarazione variabile numero di dischi
18     int nDischi;
19
20     // Presentazione gioco
21     cout << "%-----%" << endl;
22     cout << "La_Torre_di_Hanoi,_anche_conosciuta_come_Torre_di" << endl;
23     cout << "Lucas_dal_nome_del_suo_inventore,_è un rompicapo" << endl;
24     cout << "matematico_composto_da_tre_paletti_e_un_certo" << endl;
25     cout << "numero_di_dischi_di_grandezza_decrescente,_che_possono" << endl;
26     cout << "essere_infilati_in_uno_qualsiasi_dei_paletti ." << endl;
27
28     cout << endl;
29
30     // Regole del gioco
31     cout << "Regole_e_scopo." << endl;
32     cout << "Il_gioco_inizia_con_tutti_i_dischi_incolonnati_su_un" << endl;
33     cout << "paletto_in_ordine_decrescente,_in_modo_da_formare_un" << endl;
34     cout << "cono.Lo_scopo_del_gioco_è portare tutti i dischi su un" << endl;
35     cout << "paletto_diverso,_potendo_spostare_solo_un_disco_alla" << endl;
36     cout << "volta_e_potendo_mettere_un_disco_solo_su_un_altro" << endl;
37     cout << "disco_più grande, mai su uno più piccolo." << endl;
38     cout << "Fonte:_Wikipedia" << endl;
39     cout << "%-----%" << endl;
40
41     cout << endl;
42
43     // Controllo numero dischi immessi
44     // Dischi accettati: da 1 a 10
45     do {
46         cout << "Attenzione!" << endl;
47         cout << "Inserire_un_numero_di_dischi_compreso_fra_1_e_10." << endl;
48
49         cout << "Con_quanti_dischi_vuoi_giocare?_";
50         cin >> nDischi;
51     }
52     while ((nDischi < 1) || (nDischi > 11));
53
54     // Disegno i dischi
55     string disco;
56     for (int k = 0; k < nDischi; k++){
57         disco = "-OO-";

```

```

58 // Aggiungo il numero giusto di trattini
59 for(int z = 0; z < k; z++){
60     // Aggiungo un trattino all' inizio e alla fine ad ogni ciclo
61     disco = "-" + disco + "-";
62 }
63 for(int j = 1; j<=(1+nDischi-k); j++){
64     // Aggiungo la corretta spaziatura (solo a sinistra )
65     // per centrare il disco
66     disco = " " + disco;
67 }
68 cout << disco << endl;
69 }
70
71 // Disegniamo la base
72 string base = "II";
73 // Disegno la base
74 for (int i = 0; i <= nDischi; i++){
75     base = "-" + base + "-";
76     //cout << base << endl;
77 }
78 base = "|" + base + "|" + base + "|" + base + "|";
79 cout << base << endl;
80
81 // Inizio risoluzione torre Hanoi
82 cout << endl;
83 cout << "Soluzione:" << endl;
84 // Inizio cronometro
85 clock_t start = clock();
86 hanoi(nDischi, "a_sinistra", "a_destra", "al_centro");
87
88 // Fine cronometro
89 clock_t end = clock();
90 // Tempo risoluzione torre Hanoi
91 printf("Tempo_di_esecuzione_=%f_secondi\n", ((double)(end - start)) / CLOCKS_PER_SEC);
92 }
93
94 void hanoi(int n,std::string partenza,std::string arrivo,std::string ausilio){
95     if(n==1){
96         // Mossa da fare
97         spostamento(n, partenza, arrivo, ausilio );
98     }
99     else{
100         // Richiamo la funzione per spostare n-1 dischi

```

```

101 // Dal piolo partenza si arriva al piolo ausilio
102 hanoi(n-1, partenza, ausilio, arrivo);
103 // Mossa da fare
104 spostamento(n, partenza, arrivo, ausilio);
105 // Dal piolo ausilio si arriva al piolo arrivo
106 hanoi(n-1, ausilio, arrivo, partenza);
107 }
108 }
109
110 void spostamento(int n, string partenza, std::string arrivo, std::string ausilio){
111 // Stampo il numero della mossa e incremento il contatore
112 cout << "_Mossa_" << counter++ << ":\t";
113 // Quale disco devo spostare?
114 cout << "Spostare_il_disco_" << n;
115 // Da quale piolo?
116 cout << "\t_dal_piolo_" << partenza;
117 // A quale piolo?
118 cout << "\t_al_piolo_" << arrivo << "." << endl;
119 }

```

### 3. RISULTATO

---

Di seguito vengono riportate le schermate della console tre esempi di funzionamento, il primo con un disco (1 a pagina 8), il secondo con due dischi (2 a pagina 9) e l'ultimo con quattro (3 a pagina 10).

Per quanto riguarda il primo risultato, la spiegazione è ovvia. L'unica cosa che deve fare il programma per riportare la soluzione è spostare il disco dal primo piolo all'ultimo.

Il secondo risultato è un po' più elaborato. La funzione `hanoi` deve spostare due dischi. Per prima cosa confronta  $n$ , in questo caso pari a 2, con 1, ovvero la prima condizione dell'`if`. Nel caso considerato  $2 \neq 1$ , pertanto si passa all'`else` e viene richiamata la funzione `hanoi` stessa ma con un numero di dischi pari a  $n - 1 = 2 - 1 = 1$ . Ora  $1 = 1$ , quindi viene richiamata la funzione `spostamento` con `n = 1`, `partenza = "a sinistra"`, `arrivo = "al centro"` e `ausilio = "a destra"` che stampa la mossa da fare. A questo punto la seconda funzione `hanoi` termina e si ritorna alla prima continuando le istruzioni contenute nell'`else` dopo il richiamo della funzione `hanoi`. Si stampa ora la seconda mossa grazie alla funzione `spostamento` che prende in input `n = 2`, `partenza = "a sinistra"`, `arrivo = "a destra"`

e `ausilio = "al centro"`. Fatto questo viene richiamata ancora una volta la funzione `hanoi` con `n = 1` che manda in stampa l'ultima mossa.

L'ultimo caso è analogo al precedente.

Per evidenziare l'andamento non lineare del tempo di esecuzione dell'algoritmo ho effettuato una prova per ogni numero di disco accettato. I vari tempi sono riportati nella tabella 1.

Tabella 1: Tempo di esecuzione del programma.

numero di dischi	tempo di esecuzione (secondi)
1	0.000027
2	0.000037
3	0.000073
4	0.000124
5	0.000240
6	0.000488
7	0.001299
8	0.003412
9	0.004043
10	0.007189

## 4. OSSERVAZIONI E CONCLUSIONI

---

Il codice presenta alcune parti che possono essere migliorate. Il controllo del numero immesso da tastiera funziona solo con i numeri. Per esempio, se venisse scritta una qualsiasi lettera al posto di un numero, il controllo non va a buon fine e stampa all'infinito "Attenzione! Inserire un numero di dischi compreso fra 1 e 10. Con quanti dischi vuoi giocare?".

Un'altra parte del codice che può essere migliorata è il ciclo `for` per disegnare i dischi. Invece di riassegnare ad ogni iterazione la stringa `disco` e riaggiungere sempre i trattini, si potrebbe togliere gli spazi iniziali e aggiungere un solo trattino.

```

Last login: Wed Oct 26 16:03:43 on ttys002
/Users/lucazepponi/Documents/VSC/c++/Esercitazioni/Ex1/ex1 ; exit;

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
MacBook-Air-di-Luca:~ lucazepponi$ /Users/lucazepponi/Documents/VSC/c++/Esercitazioni/Ex1/ex1 ; exit;
%-----%
La Torre di Hanoi, anche conosciuta come Torre di
Lucas dal nome del suo inventore, è un rompicapo
matematico composto da tre paletti e un certo
numero di dischi di grandezza decrescente, che possono
essere infilati in uno qualsiasi dei paletti.

Regole e scopo.
Il gioco inizia con tutti i dischi incolonnati su un
paletto in ordine decrescente, in modo da formare un
cono. Lo scopo del gioco è portare tutti i dischi su un
paletto diverso, potendo spostare solo un disco alla
volta e potendo mettere un disco solo su un altro
disco più grande, mai su uno più piccolo.
%-----%

Attenzione!
Inserire un numero di dischi compreso fra 1 e 10.
Con quanti dischi vuoi giocare? 1
-00-
|--II--|--II--|--II--|

Soluzione:
- Mossa 1:      Spostare il disco 1      dal piolo A      al piolo C.
Tempo di esecuzione = 0,000016 secondi
logout

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[Processo completato]

```

Figura 1: Schermata della consolle con numero di dischi pari a uno.



```

Last login: Wed Oct 26 14:27:04 on ttys001

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
/Users/lucazepponi/Documents/VSC/c++/Esercitazioni/Ex1/ex1 ; exit;
MacBook-Air-di-Luca:~ lucazepponi$ /Users/lucazepponi/Documents/VSC/c++/Esercitazioni/Ex1/ex1 ; exit;
%-----%
La Torre di Hanoi, anche conosciuta come Torre di
Lucas dal nome del suo inventore, è un rompicapo
matematico composto da tre paletti e un certo
numero di dischi di grandezza decrescente, che possono
essere infilati in uno qualsiasi dei paletti.

Regole e scopo:
Il gioco inizia con tutti i dischi incolonnati su un
paletto in ordine decrescente, in modo da formare un
cono. Lo scopo del gioco è portare tutti i dischi su un
paletto diverso, potendo spostare solo un disco alla
volta e potendo mettere un disco solo su un altro
disco più grande, mai su uno più piccolo.
%-----%

Attenzione!
Inserire un nDischi di dischi compreso fra 1 e 10.
Con quanti dischi vuoi giocare? 2
  -00-
  --00--
|---II---|---II---|---II---|

Soluzione:
- Mossa 1:      Spostare il disco 1      dal piolo a sinistra      al piolo al centro.
- Mossa 2:      Spostare il disco 2      dal piolo a sinistra      al piolo a destra.
- Mossa 3:      Spostare il disco 1      dal piolo al centro      al piolo a destra.
Tempo di esecuzione = 0,000088 secondi
logout

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[Processo completato]

```

Figura 2: Schermata della console con numero di dischi pari a due.

```

Last login: Wed Oct 26 14:05:39 on console

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
MacBook-Air-di-Luca:~ lucazepponi$ /Users/lucazepponi/Documents/VSC/c++/Esercitazioni/Ex1/ex1 ; exit;
%-----%
La Torre di Hanoi, anche conosciuta come Torre di
Lucas dal nome del suo inventore, è un rompicapo
matematico composto da tre paletti e un certo
numero di dischi di grandezza decrescente, che possono
essere infilati in uno qualsiasi dei paletti.

Regole e scopo:
Il gioco inizia con tutti i dischi incolonnati su un
paletto in ordine decrescente, in modo da formare un
cono. Lo scopo del gioco è portare tutti i dischi su un
paletto diverso, potendo spostare solo un disco alla
volta e potendo mettere un disco solo su un altro
disco più grande, mai su uno più piccolo.
%-----%

Attenzione!
Inserire un nDischi di dischi compreso fra 1 e 10.
Con quanti dischi vuoi giocare? 4
  --00--
  ---00---
  ----00----
|-----II-----|-----II-----|-----II-----|

Soluzione:
- Mossa 1:      Spostare il disco 1      dal piolo a sinistra  al piolo al centro.
- Mossa 2:      Spostare il disco 2      dal piolo a sinistra  al piolo a destra.
- Mossa 3:      Spostare il disco 1      dal piolo al centro  al piolo a destra.
- Mossa 4:      Spostare il disco 3      dal piolo a sinistra  al piolo al centro.
- Mossa 5:      Spostare il disco 1      dal piolo a destra  al piolo a sinistra.
- Mossa 6:      Spostare il disco 2      dal piolo a destra  al piolo al centro.
- Mossa 7:      Spostare il disco 1      dal piolo a sinistra  al piolo al centro.
- Mossa 8:      Spostare il disco 4      dal piolo a sinistra  al piolo a destra.
- Mossa 9:      Spostare il disco 1      dal piolo al centro  al piolo a destra.
- Mossa 10:     Spostare il disco 2      dal piolo al centro  al piolo a sinistra.
- Mossa 11:     Spostare il disco 1      dal piolo a destra  al piolo a sinistra.
- Mossa 12:     Spostare il disco 3      dal piolo al centro  al piolo a destra.
- Mossa 13:     Spostare il disco 1      dal piolo a sinistra  al piolo al centro.
- Mossa 14:     Spostare il disco 2      dal piolo a sinistra  al piolo a destra.
- Mossa 15:     Spostare il disco 1      dal piolo al centro  al piolo a destra.
Tempo di esecuzione = 0,000138 secondi
logout

Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.

[Processo completato]

```

Figura 3: Schermata della console con numero di dischi pari a quattro.

Ho provato anche a dare una soluzione con una rappresentazione pseudo-grafica, ma ho avuto difficoltà iniziali a passare il tipo stringa come input della funzione che poi ho risolto, ma non ho più avuto abbastanza tempo per implementare questo tipo di soluzione.

## 5. RIFERIMENTI

---

Attraverso il video riportato nel seguente link, <https://www.youtube.com/watch?v=zKmIdLvx6hU>, ho approfondito l'uso delle funzioni ricorsive e ho preso spunto per la realizzazione dell'algoritmo per la risoluzione della Torre di Hanoi (ultimo accesso 24/10/2022).

Dal link [https://it.wikipedia.org/wiki/Torre\\_di\\_Hanoi#Algoritmo\\_ricorsivo](https://it.wikipedia.org/wiki/Torre_di_Hanoi#Algoritmo_ricorsivo) ho preso la spiegazione e le regole del gioco e lo schema risolutivo dell'algoritmo (ultimo accesso 24/10/2022).

Per la misurazione del tempo di esecuzione ho consultato il sito <https://clessonsonline.blogspot.com/2012/11/come-si-misura-ill-tempo-di-esecuzione.html> (ultimo accesso 24/10/2022).

Per il controllo dell'algoritmo, invece, mi sono affidato alla soluzione riportata nel PDF presente al seguente link <http://new345.altervista.org/Dispense/La%20torre%20di%20Hanoi.pdf> (ultimo accesso 24/10/2022).