

1.5em

# Statistics Exercise 02

Enrico Cotti Cottini  
Università di Padova  
Matricola 2077993

April 29, 2024

## 1 Esercizio 12 foglio 3

Consideriamo una scacchiera con 20 righe (numerata da 0 a 19 dal basso verso l'alto) e 12 colonne (numerata da 1 a 12 da sinistra a destra). Dodici pedine sono posizionate sulle caselle della riga zero. Lanciamo due dadi standard a sei facce. Spostiamo di una casella verso l'alto la pedina nella colonna il cui numero è uguale alla somma dei punteggi indicati dai due dadi. Continuiamo a lanciare i due dadi e a spostare le pedine, muovendo sempre la pedina nella colonna corrispondente alla somma dei punteggi, finché una delle pedine non raggiunge la riga 19. A questo punto, il gioco si interrompe.

Calcoliamo numericamente le probabilità dei seguenti eventi:

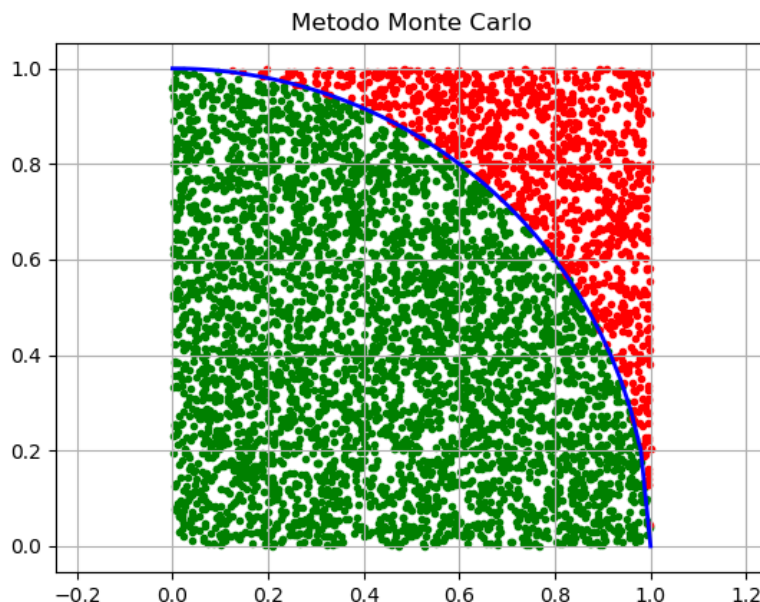
1. La pedina nella colonna sette arriva prima di quella nella colonna otto.
2. La pedina nella colonna  $k$  arriva per prima, con  $k \in \{1, \dots, 12\}$ .
3. Il gioco dura esattamente  $N$  mosse, con  $N \in \{1, \dots, 200\}$ .
4. Il gioco dura più di 200 mosse.

Per la consegna, fornire:

- Una descrizione della procedura utilizzata per calcolare le probabilità.
- Pseudocodice del programma e codice commentato in un linguaggio standard come C++ o Python (il codice anche in un file separato).
- Un grafico in formato PDF che mostri la probabilità di vittoria della pedina nella colonna  $k$  in funzione di  $k \in \{1, \dots, 12\}$ .
- Un grafico in formato PDF che mostri la probabilità che il gioco duri esattamente  $N$  mosse in funzione di  $N \in \{1, \dots, 200\}$ .

## 2 Descrizione procedura

L'approccio che ho usato è basato sull'idea del metodo Probabilistico di Montecarlo, il processo consiste nel generare un gran numero di campioni casuali secondo una distribuzione probabilistica specifica e utilizzare questi campioni per approssimare il risultato desiderato, viene utilizzato anche per il calcolo del pigreco o di integrali indefiniti. Nell'esempio del pigreco si prende un quadrato con inscritta la circonferenza goniometrica, poi iterando si sceglie casualmente un punto all'interno del quadrato, se è interno alla circonferenza lo teniamo altrimenti no, l'approssimazione del pi-greco viene ottenuta facendo il rapporto di punti estratti all'interno della circonferenza e punti totali.



Questo esempio è su  $1/4$  di circonferenza ma il ragionamento è analogo

### 2.1 applicazione al nostro esercizio

Quindi concettualmente vogliamo simulare il comportamento del gioco della scacchiera.

il gioco ha questo comportamento:

Consideriamo una scacchiera con 20 righe (numerata da 0 a 19 dal basso verso l'alto) e 12 colonne (numerata da 1 a 12 da sinistra a destra). Dodici pedine sono posizionate sulle caselle della riga zero. Lanciamo due dadi standard a sei facce. Spostiamo di una casella verso l'alto la pedina nella colonna il cui numero è uguale alla somma dei punteggi indicati dai due dadi. Quindi per valutare gli esiti della partita dovremo memorizzare:

- Se la pedina 7 è arrivata sopra alla 8 (inteso anche nei casi in cui 7 non vince la partita (potrei aver malinterpretato la consegna, in quel caso basterebbe introdurre il caso in cui 7 arriva prima di 8 ed è vincente)).
- quale pedina(colonna) vince la partita.
- quante mosse impiega la partita.
- se la partita impiega più di 200 mosse.

Quindi ad ogni simulazione della partita "estrarremo" i risultati di questi 4 punti e li rapporteremo al totale delle simulazioni.

### 3 Pseudocodice

il codice è stato sviluppato in c++ con l'utilizzo di una libreria matplotlib opensource per plottare i grafici con format simil Matlab, è presente anche una sorta di demo fatta in python che ho fatto soprattutto per capire come strutturare il codice. Consiglio di usare il codice in c++ perchè molto più rapido con numeri di simulazioni grandi. Per buildare il file main.cpp in src servono un compilatore c++ e una versione di python con intallati i moduli matplotlib e numpy.

Nella cartella src per compilare main.cpp:

Listing 1: build

```
g++ main.cpp -I/usr/include/python3.11 -lpython3.11 -o Esercizio12
```

Dove -I/usr/include/pythonX -lpythonX si riferisce alla versione globale di python che vogliamo usare dove X è la versione. Se la compilazione va a buon termine eseguire il file binario Esercizio12. oppure provare ad usare la versione demo in python in test-python-stats

#### 3.1 Pseudocodice c++

Pseudocodice della classe che rapresenta una partita:

**Classe** PartitaScacchi:

**Attributi:**

scacchiera: matrice 20x12 di interi inizializzata a 0  
posizioni pedine: vettore di 12 interi inizializzato a 0  
mosse: intero inizializzato a 0  
win: booleano inizializzato a false

**Metodo Costruttore():**

Inizializza scacchiera e posizioni pedine  
Inizializza mosse e win

**Metodo** lancioDadi():

Restituisce la somma di due dadi a sei facce

**Metodo** avanzaPedina(colonna):

Sposta la pedina nella colonna data in alto di una cella

**Metodo** check win():  
 Restituisce true se una pedina raggiunge la riga 19

**Metodo** winning column():  
 Restituisce la colonna in cui è stata vinta la partita

**Metodo** pedina colonna 7 prima di 8():  
 Restituisce true se la pedina nella colonna 7 arriva prima di quella nella colonna 8

**Metodo** pedina colonna k prima di tutte le altre(k):  
 Restituisce true se la pedina nella colonna k arriva prima di tutte le altre

**Metodo** check mosse 200():  
 Restituisce true se il numero di mosse è compreso tra 0 e 200

**Metodo** results():  
 Stampa i risultati della partita

**Metodo** play():  
 Simula la partita finché non si verifica una condizione di vittoria

**Metodo** reset():  
 Reimposta lo stato della partita

Il resto del codice serve: iterare tante simulazioni, memorizzare i risultati, fare il rapporto tra i casi giusti e quelli totali per calcolarne le probabilità, scrivere i risultati in scacchiera.txt, plottare i grafici e salvarli in pdf.

## 4 Grafici

Nel primo ho un grafco in formato pdf che riporti la probabilità di vittoria della pedina. anche presente in vittoria.pdf

in colonna k in funzione di k 1, . . . , 12; Nel secondo un grafco in formato pdf che riporti la probabilità che il gioco abbia durata di esattamente N mosse in funzione di N 1, . . . , 200. anche presente in mosse.pdf

Inolte il file statistiche.txt contiene tutte le informazioni piu nel dettaglio dei risultati della simulazione.

