Laboratorio di Calcolo, Esercitazione 6, 17-21 novembre 2025

Canale Pet-Z, Docenti: Shahram Rahatlou, Fabio Bellini, Sibilla Di Pace

Cammino aleatorio unidimensionale: Lo scopo di questa esercitazione è di studiare il cammino aleatorio utilizzando numeri casuali tramite le funzioni `srand48()` es `lrand48()` ed immagazzinando i dati in array.

Il cammino aleatorio (random walk) unidimensionale descrive il moto una particella lungo una retta a partire da una posizione iniziale x_0. Ad ogni passo, la particella ha la stessa probabilità di avanzare (di una cella) a destra o a sinistra. La simulazione finisce dopo N_{step} passi.

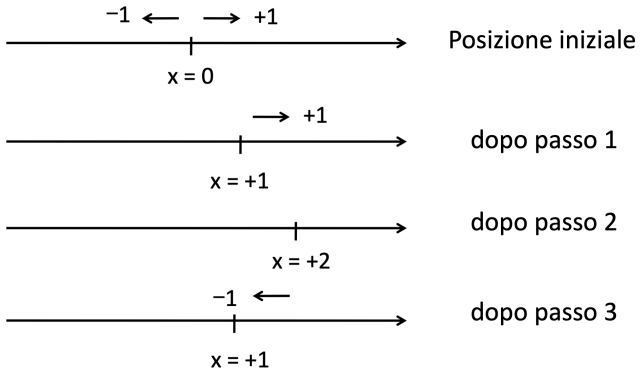

Figura 1: Esempio del cammino aleatorio dopo 3 passi

In questa prova si vogliono studiare alcune caratteristiche del cammino aleatorio al variare del numero di passi N_{step} per $N_{\text{exp}} = 10000$ simulazioni. Ciascun cammino finisce dopo N_{step} passi in una posizione finale `posFinal` nell'intervallo $[-N_{\text{step}}, +N_{\text{step}}]$.

► Cartella di lavoro

Fare login sulla postazione utilizzando le credenziali del vostro gruppo, `lcsrNNN`, dove `NNN` è il vostro numero di gruppo, ad esempio `098`. Creare una cartella `LCSR6` nella *home directory* con il comando `mkdir` in cui scriverei programmi di oggi. Spostarsi in questa cartella con il comando `cd`. Tutti i file di codice sorgente in C e in python dovranno trovarsi in questa cartella per essere visualizzati. **Le cartelle create sulla scrivania (Desktop) o in altre sotto-cartelle non verranno copiate né valutate.**

Prima parte

Creare un programma `walk.c`, dove `NNN` è il vostro numero di gruppo, ad esempio `098`, nella cartella `LCSR6` utilizzando l'editor di testo `emacs`, per eseguire le seguenti operazioni:

1. Fare $N_{\text{exp}} = 10000$ simulazioni del cammino aleatorio a partire dall'origine con $N_{\text{step}} = 20$;
2. per ciascuna simulazione calcolare
   - la posizione più lontana raggiunta durante il cammino (`maxDist`)
   - la posizione finale del cammino (`posFinal`);

Salvare questi dati per ciascuna simulazione in array di opportuna lunghezza.

3. ogni 500 simulazioni stampare sullo schermo il numero di simulazione e la posizione finale raggiunta al termine del cammino;

4. per ciascuna posizione finale possibile, memorizzare in un array `frequenza` di lunghezza opportuna, il numero di simulazioni terminate in quella posizione;

5. Al termine delle simulazioni

   - stampare sullo schermo la media aritmetica su tutte le simulazioni della distanza della posizione finale dall'origine (`distFinMedia`)
   - stampare sullo schermo un sommario dei valori immagazzinati nell'array `frequenza` con il seguente formato

```
****************** Sommario posizioni finali ****************
posizione: -20   simulazioni:     0
posizione: -18   simulazioni:     0
posizione: -16   simulazioni:     1
posizione: -14   simulazioni:    14
posizione: -12   simulazioni:    37
posizione: -10   simulazioni:   156
posizione:  -8   simulazioni:   384
posizione:  -6   simulazioni:   759
posizione:  -4   simulazioni:  1158
posizione:  -2   simulazioni:  1606
posizione:  +0   simulazioni:  1724
posizione:  +2   simulazioni:  1667
posizione:  +4   simulazioni:  1212
posizione:  +6   simulazioni:   725
posizione:  +8   simulazioni:   377
posizione: +10   simulazioni:   125
posizione: +12   simulazioni:    46
posizione: +14   simulazioni:     6
posizione: +16   simulazioni:     3
posizione: +18   simulazioni:     0
posizione: +20   simulazioni:     0
```

6. usando le funzioni `fopen()` e `fprintf()` scrivere questi dati, posizione finale e la sua frequenza (solo 2 numeri per riga), in un file `frequenza.txt` con il formato `"%+5d %10d"`;

7. scrivere un programma python `frequenza.py` per caricare i dati dal file `frequenza.txt` e produrre il grafico a barra che mostra per ciascuna posizione finale il numero di simulazioni terminate in quella posizione;

Si ricorda che per creare l'eseguibile utilizzando la libreria matematica dovete usare il comando `gcc -Wall -o app.exe programma.c -lm` dalla riga di comando nel terminale.

Si consiglia di scrivere il programma in modo incrementale, verificando la corretta compilazione e l'esecuzione almeno dopo ciascuno dei passi indicati nel testo.

```python
# carica moduli pyplot, numpy e matematica di python
import matplotlib.pyplot as plt
import numpy as np
import math as m

##
####  istogramma della distribuzione num simulazioni in funzion della
   posizione finale
##
pos, freq = np.loadtxt('frequenza.txt', unpack=True)

plt.title("Distribuzione posizioni finali con $N_{passi} = 20$")

# istogramma dei valori letti dal file
plt.bar( pos, freq, color='blue', label='posizione finale randomwalk')

plt.xlabel("coordinata posizione finale")
plt.ylabel("# simulazioni")

# linea vertical a x = 0
plt.axvline( x=0., color = 'red', linestyle='--')
plt.legend()
plt.show()
```

Listato 1: Programma `frequenza.py`

► **Seconda parte**

In questa seconda parte, vogliamo studiare l'andamento della distanza media finale in funzione del numero di passi del cammino, variando N_{step}. A tal fine, dobbiamo creare un file di dati `distfin.txt` che contenga per ciascuna riga solo due numeri

1. il numero di passi del cammino N_{step};
2. il valore di `distFinMedia`;

Scrivere un nuovo programma `walk2.c` modificando opportunamente il vostro programma `walk.c` nel modo seguente:

1. con un opportuno ciclo variare il numero N_{step} aumentandolo di un fattore 2, $N_{\text{step}} = 2, 4, 8, 16, 32, 64, 128$
2. per ciascun valore di N_{step} riptere il cammino aleatorio (quanto fatto nella prima parte) commentando la stampa sullo schermo della prima parte;
3. al termine delle simulazioni per ciascun valore di N_{step}, stampare sullo schermo il valore di N_{step} e di `distFinMedia`;
4. scrivere N_{step} e `distFinMedia` nel file `distfin.txt` con il formato `"%5d %.1f"`
5. scrivere il codice python `andamento.py` per graficare l'andamento della distanza media finale in funzione del numero di passi N_{step}.

► **Nozioni utili**

1. l'inizializzazione dei numeri casuali con la funzione `srand48( seed )` va fatta una sola volta e all'inizio della funzione `main()`;
2. per N valori $\{x_1, ..., x_N\}$, la media aritmetica $\langle x \rangle$ è definita come $\langle x \rangle = \sum_{j=1}^{N} x_j / N$
3. per eseguire il codice in python, dalla riga di comando nel terminale dovete eseguire il comando `python3 nomefile.py`