

Laboratorio di Calcolo, Prova d'esame del 15/09/2025

Tutti i canali, Anno accademico 2024-25

Nome: _____ Cognome: _____
Matricola: _____ ☐ Ritirata/o

Lo scopo di questa prova d'esame è di scrivere un programma in C e uno script in python seguendo la traccia riportata di seguito. Si tenga presente che:

1. Per svolgere il compito si hanno a disposizione 3 ore.
2. Si possono usare libri di testo, prontuari e gli appunti ma non è ammesso parlare con nessuno né utilizzare cellulari, tablet o laptop, pena l'annullamento del compito.
3. **Tutti i file vanno salvati in una cartella chiamata LCSEP25_NOME_COGNOME nella home directory**, dove NOME e COGNOME indicano rispettivamente il vostro nome e cognome. Ad esempio lo studente *Nicolò Maria De Rossi Salò* deve creare una cartella chiamata LCSEP25_NICOLOMARIA_DEROSSISALO contenente tutti i file specificati nel testo. **Tutto ciò che non si trova all'interno della cartella suddetta non verrà valutato.** In tutti i programmi e script inserite all'inizio un commento con il vostro nome, cognome e numero di matricola.
4. **Dovete consegnare il presente testo indicando nome, cognome e numero di matricola** (vedi sopra), barrando la casella "Ritirato/a" se ci si vuole ritirare, ovvero se non si vuole che la presente prova venga valutata.
5. **Per consegnare il compito** dovreste eseguire, all'interno della cartella creata in precedenza (come spiegato al punto 3), il seguente comando da terminale: `cp * /media/sf_esame/`

Consideriamo una variabile casuale discreta che può assumere valori **interi** $n = 0, \dots, N-1$, distribuiti secondo una distribuzione di probabilità discreta $p(n)$. Il metodo di Metropolis-Hastings descritto in questo esercizio permette di generare una serie di numeri casuali appartenenti a una $p(n)$ qualsiasi, utilizzando le cosiddette *catene di Markov*. In questo esercizio per considereremo una distribuzione di probabilità *random*, in cui ad ogni numero m è associata una probabilità $p(m)$ casuale, tale che $\sum_{m=0}^{N-1} p(m) = 1$.

L'idea di base è quella di costruire una sequenza di numeri interi n_k , seguendo una procedura tale che, dopo un certo numero di passi k , gli elementi della sequenza seguano la distribuzione desiderata.

Una possibile implementazione del metodo di Metropolis-Hastings è la seguente:

1. Si fissa il numero totale di elementi della sequenza `NSAMPLE`.
2. Si genera un numero casuale n_0 , compreso tra 0 e $N-1$, che sarà l'elemento iniziale della sequenza.
3. A partire da n_0 si costruiscono gli elementi n_k della sequenza, ripetendo le seguenti operazioni per $0 \leq k \leq (\text{NSAMPLE}-1)$:
 - (a) Si genera casualmente un numero intero "di prova" n_{tmp} , compreso tra 0 e $N-1$;

(b) Si calcola il rapporto

$$A = \min\left(1, \frac{p(n_{\text{tmp}})}{p(n_k)}\right),$$

dove $p(n)$ è la distribuzione di probabilità richiesta.

(c) Si estrae casualmente un numero reale y con distribuzione uniforme tra 0 e 1.

- Se $y \leq A$, il numero di prova viene accettato, e si pone $n_{k+1} = n_{\text{tmp}}$.
- Se $y > A$, il numero proposto viene rifiutato e si pone $n_{k+1} = n_k$.

In questo modo si costruisce una sequenza di valori n_k ; per convenzione si esclude il valore iniziale n_0 , e si considerano come appartenenti alla sequenza solo gli `NSAMPLE` valori $n_1, n_2, \dots, n_{\text{SAMPLE}}$.

Se `NSAMPLE` è sufficientemente grande, la sequenza finale di numeri avrà una distribuzione prossima a $p(n)$, come desiderato.

► **Esercizio in C:** Scrivere un programma `NOME.COGNOME.c` che implementi il metodo di Metropolis-Hastings, rispettando le seguenti specifiche:

1. Definire, tramite opportune direttive `define`, (i) il numero di elementi della sequenza `NSAMPLE` = 10000; il numero di possibili valori della variabile casuale `N` = 10;
2. Contenere tre array, di tipo e dimensioni opportune: (i) `p[]` per memorizzare la distribuzione di probabilità; (ii) `samples[]` per contenere i valori della sequenza generata dal metodo di Metropolis-Hastings; (iii) `freq[]` per contenere la distribuzione approssimata generata dal metodo di Metropolis-Hastings. Si noti che questo implica che `freq[]` e `p[]` dovranno avere la stessa dimensione e tipo.
3. Contenere una funzione `init_prob()`, di tipo e argomenti opportuni, che inizializzi l'array `p[]` come segue:
 - 3a. Riempire l'array `p[]` con `N` numeri casuali (di tipo `double`), estratti con distribuzione uniforme nell'intervallo $[0, 1]$;
 - 3b. Calcolare la somma di tutti gli elementi di `p[]`, ottenendo così la costante di normalizzazione della distribuzione `norm`;
 - 3c. Normalizzare quindi la distribuzione dividendo ogni elemento di `p[]` per `norm`.
4. Contenere la funzione `metropolis()`, di tipo e argomenti opportuni, che riempia l'array `samples[]` con gli `NSAMPLE` valori generati usando l'algoritmo di Metropolis-Hastings descritto precedentemente.
5. Contenere inoltre la funzione `frequency()`, di tipo e argomenti opportuni, che riempia l'array `freq[]` con la frequenza con cui ciascun valore possibile di n , con $0 \leq i \leq N-1$, compare nella sequenza contenuta nell'array `samples[]`. Si ricorda che per ottenere la frequenza bisogna dividere il numero di occorrenze di ciascun valore di n per il numero totale di elementi generati (`NSAMPLE`).
6. Una volta generata la sequenza tramite la funzione `metropolis()`, e calcolata la distribuzione di probabilità approssimata tramite la funzione `frequency()`, si crei un file di tre colonne di nome `frequency.dat`, che contenga i valori possibili di n , e i valori esatti (`p[]`) e approssimati (`freq[]`) della distribuzione, nel formato:

```
0 p(0) freq(0)
1 p(1) freq(1)
...
N-1 p(N-1) freq(N-1)
```

I valori della seconda e terza colonna devono essere stampati con 4 cifre decimali.

► **Esercizio in Python:**

Dopo aver verificato che il programma funzioni correttamente, creare uno script python `NOME_COGNOME.py` che legga i dati contenuti nel file `frequency.dat` e riporti su un unico grafico in maniera leggibile i valori di $p(n)$ e $\text{freq}(n)$ in funzione di n , salvandolo sul file `NOME_COGNOME.png`. Il grafico deve essere completo di titolo e di opportune legende per i dati e gli assi.