

# Esame di Programmazione Informatica

16 giugno 2021

## Informazioni generali

- 2 esercizi di script MATLAB e 2 domande a risposta multipla.
- La durata dell'esame è di 2 ore.
- È consigliato l'uso di MATLAB nello svolgimento di tutti i punti.
- Le slide e gli script del corso sono consultabili liberamente.
- Il punteggio assegnato ad ogni esercizio ed alle domande a risposta multipla è indicato tra parentesi.

## Esercizio 1 (14/30)

Scrivere una funzione MATLAB che prenda in ingresso:

- il *function handle* di una funzione  $f$ ,
- un numero reale  $x$ ,
- un vettore colonna  $\mathbf{v} = [v_1; \dots v_i \dots; v_N]$  di lunghezza  $N$

e restituisca la frazione  $P$  di elementi di  $\mathbf{v}$  che rispettano la condizione  $f(v_i) < x$ ,  $i = 1, \dots, N$ .

$P$  si calcola quindi contando il numero di volte  $n$  che la precedente condizione viene rispettata per gli  $N$  elementi  $v_i$  del vettore  $\mathbf{v}$ , dividendo poi per la lunghezza  $N$  del vettore  $\mathbf{v}$ , cioè  $P = n/N$ . Utilizzare preferibilmente istruzioni vettoriali (senza cicli `for`).

Si mostri poi come utilizzare la funzione scritta precedentemente per calcolare la frazione  $P$  nel caso di:

- funzione  $f(t) = t^2$  definita mediante funzione anonima,
- un numero reale  $x = 0.5$ ,
- un vettore colonna  $\mathbf{v}$  di  $N = 10^6$  numeri casuali con distribuzione uniforme nell'intervallo  $[0, 1]$ , utilizzando cioè la seguente istruzione MATLAB:

`$\mathbf{v} = \text{rand}(N, 1)$`

e si confronti il risultato ottenuto con quello teorico  $\bar{P} = \sqrt{0.5}$ .

*Soluzione:* assumendo che la funzione  $f$  agisca in maniera vettoriale (input e output di  $f$  sono vettori), continuiamo ad operare in forma vettoriale per maggiore comodità, compattezza ed efficienza.

Calcoliamo quindi il vettore booleano (di tipo `logical`)  $f(v) < x$  che indica, componente per componente, per quali valori del vettore  $v$  la condizione  $f(v_i) < x$  viene rispettata. Per contare quanti di questi elementi sono veri, sfruttiamo la funzione somma vettoriale `sum` che considererà 1 i booleani veri (`true`) e 0 quelli falsi (`false`).

Dividiamo infine per la lunghezza  $N$  del vettore  $v$ , ottenuto con la funzione `length(v)`:

`conteggio.m`

```
function P = conteggio(f, v, x)
    P = sum( f(v) < x ) / length(v) ;
end
```

Utilizzo della funzione scritta precedentemente per calcolare la frazione  $P$  nel caso di  $f(t) = t^2$ , definita mediante funzione anonima con elevamento al quadrato elemento per elemento (`.^ 2`),  $x = 0.5$  e vettore  $v$  random lungo  $N = 10^6$ :

```
f = @(t) t .^ 2 ;
x = 0.5 ;
N = 1e6 ;
v = rand(N, 1) ;

P = conteggio(f, v, x)
P_teorico = sqrt(x)
```

ottenendo correttamente  $P \approx P_{\text{teorico}} = 0.7071\dots$

## Esercizio 2 (14/30)

Si consideri la successione

$$x_i = x_{i-1} + x_{i-2}$$

con valori iniziali  $x_0 = 0$  e  $x_1 = 1$ , i cui primi termini valgono quindi

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, \dots$$

Scrivere una funzione MATLAB che prenda in ingresso un valore  $X$  e restituisca il più piccolo indice  $i$  della successione per il quale  $x_i \geq X$ . Operativamente è necessario calcolare esplicitamente la successione termine a termine, partendo dai precedenti valori iniziali e fermandosi non appena l'ultimo termine della successione che viene calcolato supera, o è uguale, al valore  $X$  dato.

Si mostri poi come utilizzare la precedente funzione nei casi  $X = 10, 100, 1000, 10000$  ed i relativi risultati.

*Soluzione:* calcoliamo esplicitamente la successione termine a termine partendo dai valori iniziali  $x_0 = 0$  e  $x_1 = 1$  utilizzando un ciclo **while** per controllare l'arresto del calcolo non appena la condizione richiesta, ossia  $x_i \geq X$ , viene rispettata. La condizione di controllo del ciclo **while** sarà quindi la complementare  $x_i < X$ . Nel caso non si voglia tenere traccia in memoria di tutti i valori calcolati della successione in un vettore, sarà necessario copiare i nuovi valori nei vecchi ad ogni iterazione del ciclo **while**:

termine\_successione.m

```
function i = termine_successione(X)
    x0 = 0 ;
    x1 = 1 ;
    i = 1 ;
    xi = 1 ;
    while xi < X
        i = i + 1 ;
        xi = x0 + x1 ;
        x0 = x1 ;
        x1 = xi ;
    end
end
```

Utilizzo della precedente funzione nei casi  $X = 10, 100, 1000, 10000$ , per comodità usiamo un ciclo **for** per considerare automaticamente i 5 valori di  $X$  richiesti:

```
for e = 1 : 5
    X = 10^e ;
    i = termine_successione(X)
end
```

ottenendo correttamente  $i = 7, 12, 17, 21, 26$ .

## Domande a risposta multipla (5/30)

### Domanda 1 (2/30)

Data la funzione  $f(x) = x - 3x^3$ , è corretto cercare i suoi punti di minimo e di massimo relativi con il seguente script?

```
x = linspace(-1, 1, 1000) ;  
f = x - 3*x .^ 3 ;  
min_f = min(f) ;  
max_f = max(f) ;
```

- sì, perchè i massimi ed i minimi relativi coincidono con quelli assoluti.
- sì, perchè vi sono esattamente un massimo ed un minimo.
- no, perchè si otterrebbero gli estremi in quel particolare intervallo.
- no, perchè la funzione non ha estremi nè relativi nè assoluti.

*Soluzione:* no, perchè si otterrebbero gli estremi in quel particolare intervallo.

### Domanda 2 (3/30)

Data la seguente funzione MATLAB:

```
function y = f(x)  
    y = x ;  
    if x > 0  
        x = 1 ;  
    else  
        x = -1 ;  
    end  
end
```

che funzione matematica  $f(x)$  essa implementa?

- $f(x) = |x|$
- $f(x) = x/|x| = \text{sgn } x$  (funzione segno, 0 escluso)
- $f(x) = x$
- $f(x) = -|x|$

*Soluzione:*  $f(x) = x$ .