

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Linear Congruential Generator

Esercizio: Linear Congruential Generator

Si consideri la successione (linear congruential generator):

$$X_{n+1} = (aX_n + c) \bmod m$$

- Con X_i , a , c ed m interi; il valore di partenza X_0 è noto

Nel file di funzione `es_lcg.m` si definisca la funzione ausiliaria:

```
function R = my_lcg(X0, N)
```

- Che, dato il valore (scalare) di X_0 ...
- ...Generi gli elementi da X_1 a X_N della successione...
- ...E li salvi nel vettore di ritorno R

Si consideri $m = 16$, $a = 9$, $c = 3$

Esercizio: Linear Congruential Generator

Qualche suggerimento:

- Prima concentratevi sull'implementare l'iterazione fondamentale:

$$X_{n+1} = (aX_n + c) \bmod m$$

- Quindi pensate a salvare i valori nel vettore di uscita

Si verifichi la correttezza nella funzione principale `es_lcg`:

- Si provi ad eseguire `my_lcg` in sequenza con diversi valori di `x0`
- Come variano i valori di uscita?

Esercizio: Linear Congruential Generator

Noterete che i valori di uscita variano **in modo imprevedibile**

- LCG può essere utilizzato per ottenere **numeri pseudo-casuali**...
- Ossia numeri che **sembrano casuali**, ma **non lo sono**
- La sequenza è controllata da X_0 , che si dice anche "seme" (**seed**)

La funzione **rand** in Matlab funziona in modo simile! Ci ritorneremo su...

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Trovare Elementi
in un Vettore

Esercizio: Trovare Elementi in un Vettore

Nel file di funzione `es_findidx.m`, definire la funzione ausiliaria:

```
function I = my_findidx(X, V)
```

- Con **x** scalare e **v** vettore, che restituisca nel vettore **I**...
- ...Gli indici degli elementi che in **v** sono uguali a **x**

Per esempio:

```
V = [1, 2, 4, 2];  
my_findidx(2, V) % Denota [2, 4]
```

- Si implementi la funzione utilizzando un ciclo **for**

Esercizio: Trovare Elementi in un Vettore

Si verifichi la correttezza nella funzione principale `es_findidx`:

- Si utilizzino dei vettori di numeri interi casuali, ottenuti con `randi`

```
function R = randi(IMAX, N)
function R = randi(IMAX, M, N)
```

- Funziona come `rand`, ma genera *interi* tra 1 e `IMAX`

Come ottenere lo stesso risultato con le funzioni predefinite?

- Si determini un metodo e lo si sfrutti per fare un confronto
- Suggerimento: vi ricordate della funzione `find`?

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Scambio Semplice

Esercizio: Scambio Semplice

Nel file di funzione `es_swap.m`, definite la funzione ausiliaria:

```
function W = my_swap(V, I, J)
```

Con \mathbf{v} vettore, \mathbf{I} e \mathbf{J} indici. La funzione:

- Deve restituire un vettore \mathbf{w} , identico a \mathbf{v} ...
- ...Ma con gli elementi in posizione \mathbf{I} e \mathbf{J} scambiati

Verificate la correttezza nella funzione principale `es_swap`:

- Utilizzate vettori casuali o inseriti a mano
- Controllate il risultato visivamente

Suggerimento: stavolta non serve iterare!

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Scorrimento Circolare

Esercizio: Scorrimento Circolare

Nel file di funzione `es_shift.m`, definite la funzione ausiliaria:

```
function W = my_shift(V)
```

- Deve restituire un vettore \mathbf{w} , identico a \mathbf{v} ...
- ...Ma con gli elementi spostati a sx di una posizione...
- ...E in modo circolare: il primo elemento diventa l'ultimo

Per esempio:

```
my_shift([1, 2, 3, 4]) % Denota [2, 3, 4, 1]
```

Esercizio: Scorrimento Circolare

Verificate la correttezza nella funzione principale `es_swap`:

- Utilizzate vettori casuali o inseriti a mano
- Controllate il risultato visivamente

Suggerimento:

- Si può implementare la funzione scambiando gli elementi consecutivi
- ...Ossia scambiando `w(ii)` con `w(ii+1)` in sequenza

Elementi di Informatica e Applicazioni Numeriche T

Esercizio: Bubble Sort

Esercizio: Bubble Sort

"Bubble sort" un algoritmo semplice per ordinare un vettore

- L'algoritmo **scorre ripetutamente un vettore**
- Ad ogni passaggio, si **scambiano gli elementi consecutivi fuori ordine**
- Si procede finché **non sono più necessari scambi**

Ricordate l'esempio dell'ordinare l'aula nella prima lezione?

- Bubble sort è l'algoritmo che avevamo pensato per risolverlo!

Esercizio: Bubble Sort

Lo pseudo-codice di bubble sort è:

```
ordinato = false
while ordinato = false
  ordinato = true (assumo che il vettore sia ordinato)
  for  $i = 1..n - 1$  ( $n$  è la lunghezza, mi fermo un passo prima)
    if  $V_i > V_{i+1}$ 
      scambia gli elementi
      ordinato = false
    end
  end
end
end
```

Esercizio: Bubble Sort

Nel file di funzione `es_bubble.m`, definite la funzione ausiliaria:

```
function W = my_bubble(V)
```

- Che restituisca il vettore \mathbf{w} , con gli stessi elementi di \mathbf{v} ...
- ...ma ordinati utilizzando bubble sort.

Verificate la correttezza nella funzione principale `es_bubble`:

- Utilizzate vettori casuali o inseriti a mano
- Confrontate il risultato con quello della funzione predefinita `sort`:

```
function W = sort(V)
```