



Informatica B

2017-2018

Esercitazione VI

Ricorsione (2)

Alessandro A. Nacci

alessandro.nacci@polimi.it - www.alessandronacci.it



Esercizio I

Scrivere una funzione ricorsiva per il calcolo approssimato del numero di eulero e . Il numero e può essere ottenuto come :

$$e := \sum_{n=0}^{\infty} \frac{1}{n!} = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots$$

La sommatoria converge ad e (costante di Eulero) ed il suo calcolo ad un determinato passo restituisce un'approssimazione più o meno buona del numero e . La funzione richiesta prende quindi in ingresso il numero di passi della sommatoria da simulare e restituisce l'approssimazione di e a quel passo.

```
function [ eN ] = es5( N )  
  
    if (N==0)  
        eN = 1/factorial(0);  
    else  
        eN = 1/factorial(N) + es5(N-1);  
    end  
  
end
```

Nel gioco della roulette si ha una ruota con 37 numeri da 0 a 36.

- Si scriva una funzione che simuli la ruota e generi il numero vincente accettando come parametro un generatore di numeri casuali compresi tra 0 e 36.
- Si scriva una funzione che provi a eseguire N giri e salvi i risultati delle frequenze di ogni numero in un array di 37 elementi (il primo elemento per lo 0, il secondo per l'1 e così via). La funzione riceverà in ingresso anche il generatore di numeri casuali.
- Si supponga di ripetere il calcolo precedente con una roulette truccata che ha probabilità doppia di far uscire uno 0 rispetto a qualsiasi altro numero.
- Rappresentare con un grafico l'array dei 37 elementi e calcolare la probabilità del risultato





Esercizio 2

Nel gioco della roulette si ha una ruota con 37 numeri da 0 a 36.

- Si scriva una funzione che simuli la ruota e generi il numero vincente accettando come parametro un generatore di numeri casuali compresi tra 0 e 36.

```
function [ num ] = es4_estrazione( generatore )
```

```
    num = generatore();
```

```
end
```

```
function [ num ] = generatore()
```

```
    num = randi(37) - 1;
```

```
end
```

$X = \text{randi}(\text{imax})$ returns a pseudorandom scalar integer between 1 and imax.



Esercizio 2

- Si scriva una funzione che provi a eseguire N giri e salvi i risultati delle frequenze di ogni numero in un array di 37 elementi (il primo elemento per lo 0, il secondo per l'1 e così via). La funzione riceverà in ingresso anche il generatore di numeri casuali.

```
function [ result ] = es4_tutteEstrazioni( N, generatore )  
  
    result = zeros(1,37);  
  
    for i = 1:N  
        num = generatore();  
        result(num+1) = result(num+1) + 1;  
    end  
  
end
```



Esercizio 2

- Si supponga di ripetere il calcolo precedente con una roulette truccata che ha probabilità doppia di far uscire uno 0 rispetto a qualsiasi altro numero.

```
function [ num ] = generatore(T)

    n = rand();

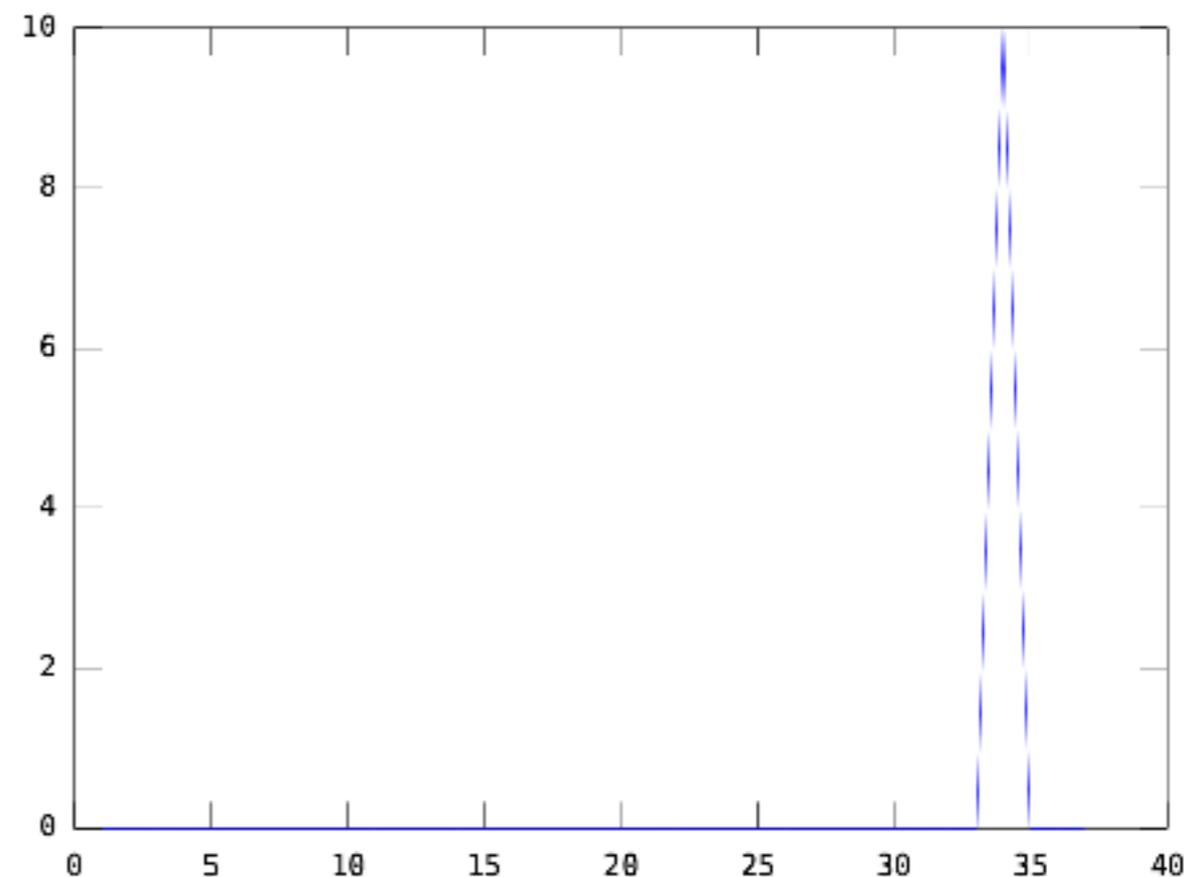
    if(n> T )
        num = 0;
    else
        num = generatoreFair();
    end
```



Esercizio 2

- Rappresentare con un grafico l'array dei 37 elementi e calcolare la probabilità del risultato

```
function [ result ] = es4_calcoloProbabilita( N, generatore )  
  
    result = zeros(1,37);  
  
    for i = 1:N  
        num = generatore();  
        result(num+1) = result(num+1) + 1;  
    end  
  
    x_axes = [1:37];  
  
    plot(x_axes, result);  
  
    result = result / N;  
  
end
```





Simulazione Prova d'Esame



Esercizio I

Esercizio 1 (6 punti)

Scrivere una funzione non ricorsiva di nome **frame** che riceve un parametro **N** (intero pari positivo) e ritorna una matrice **A** di dimensione $N \times N$ in cui:

- gli elementi della riga 1, riga N, colonna 1 e colonna N sono inizializzati a 1
- gli elementi **rimanenti** della riga 2, riga N-1, colonna 2 e colonna N-1 sono inizializzati a 2
- gli elementi **rimanenti** della riga 3, riga N-2, colonna 3 e colonna N-2 sono inizializzati a 3
- così via fino a completare tutta la matrice

Ad esempio, per $N = 8$:

```
> frame(8)
```

```
ans =
```

```
1 1 1 1 1 1 1 1
1 2 2 2 2 2 2 1
1 2 3 3 3 3 2 1
1 2 3 4 4 3 2 1
1 2 3 4 4 3 2 1
1 2 3 3 3 3 2 1
1 2 2 2 2 2 2 1
1 1 1 1 1 1 1 1
```

Nota: il massimo livello di annidamento tra istruzioni iterative (es. **for**) innestate utilizzabili per implementare la funzione è 2.



Esercizio I

Punto facoltativo: modificare la funzione **frame**, senza aggiungere ulteriori cicli, in modo tale che gli elementi appartenenti alle due diagonali della matrice siano uguali a -1; ad esempio:

```
> frame(8)
```

```
ans =
```

```
-1  1  1  1  1  1  1  -1  
 1 -1  2  2  2  2 -1  1  
 1  2 -1  3  3 -1  2  1  
 1  2  3 -1 -1  3  2  1  
 1  2  3 -1 -1  3  2  1  
 1  2 -1  3  3 -1  2  1  
 1 -1  2  2  2  2 -1  1  
-1  1  1  1  1  1  1  -1
```



Esercizio I

Soluzione

Sono possibili varie soluzioni. Quella che viene proposta qui sotto si basa sull'utilizzo dei sotto-array. In particolare, nella soluzione si identificano come sottoarray tutte le "cornici" concentriche che sono ricavabili dalla matrice, partendo dalla cornice più esterna, che è costituita dalla prima e ultima riga e dalla prima e ultima colonna e proseguendo verso il centro della matrice stessa. Agli elementi di ogni cornice vengono assegnati valori crescenti, da 1 (assegnato a tutti i valori della cornice esterna) a $N/2$ (assegnato agli elementi della "cornice" più interna).

```
function [A] = frame(N)
    for ii=1:N/2
        A([ii,N-ii+1],ii:N-ii+1) = ii;
        A(ii:N-ii+1,[ii,N-ii+1]) = ii;
    end
```

Parte facoltativa: si aggiunge un'istruzione che, dopo aver selezionato un sotto-array composto da tutti gli elementi che si trovano in una delle due diagonali, assegna a tali elementi il valore -1.

```
function [A] = frame(N)
    for ii=1:N/2
        A([ii,N-ii+1],ii:N-ii+1) = ii;
        A(ii:N-ii+1,[ii,N-ii+1]) = ii;
        A([ii,N-ii+1],[ii,N-ii+1]) = -1;
    end
```



Esercizio 2

Esercizio 2 (7 punti)

Si considerino la funzione e lo script seguenti:

```
function [r] = MiaFunz(a, b)
    if b == 0
        r = 1;
    elseif b < 0
        r = a * MiaFunz(a, b+1);
    else
        r = 1/a * MiaFunz(a, b-1);
    end;
```

```
%script che chiama MiaFunz
x = 2;
for y = -3:1:3
    r = MiaFunz(x,y)
end
```

1. Descrivere brevemente quale sia la funzione matematica calcolata dalla funzione MATLAB MiaFunz. Giustificare la risposta.
2. Quali risultati vengono stampati a video? Non è necessario calcolare i valori numerici esatti, ma è sufficiente riportare le espressioni aritmetiche necessarie per calcolarli. Giustificare la risposta.



Esercizio 2

1. La funzione matematica calcolata dal programma è a^{-b} tranne che nel caso in cui sia a che b sono uguali a zero in cui viene restituito il valore 1.

Giustificazione

- Nel caso base in cui b è uguale a zero, la funzione restituisce sempre il valore 1.
- Nel caso in cui b è negativo, la funzione restituisce il valore di a moltiplicato per il risultato della funzione stessa chiamata ricorsivamente con parametri a e $b+1$. Questa chiamata ricorsiva restituisce il valore di a moltiplicato per il risultato della successiva chiamata ricorsiva ottenuto con stesso a e valore di b incrementato di un'ulteriore unità. La ricorsione termina quando b raggiunge il valore 0. Quindi si hanno esattamente $|b|$ ($-b$ considerando che b è negativo) passi ricorsivi in cui il valore a è moltiplicato per se stesso.

Nel caso generale si ha perciò la seguente situazione (ricordando che in questo caso il valore di b iniziale è negativo e viene incrementato ad ogni passo fino ad arrivare a zero):

$$\begin{aligned} \text{MiaFunz}(a,b) &= \\ (\text{passo } 1) &= a \text{ MiaFunz}(a,b+1)= \\ (\text{passo } 2) &= a^2 * \text{MiaFunz}(a,b+2)= \\ (\text{passo } 3) &= a^3 * \text{MiaFunz}(a,b+3)=... \\ (\text{passo } -b-2) &= a^{-b-2} * \text{MiaFunz}(a,b+(-b-2))= \\ (\text{passo } -b-1) &= a^{-b-1} * \text{MiaFunz}(a,b+(-b-1))= \\ (\text{passo } -b) &= a^{-b} * \text{MiaFunz}(a,b+(-b))= \\ &= a^{-b} \end{aligned}$$

- Nel caso in cui b è positivo, la funzione si comporta in modo simile al caso precedente, ma questa volta il valore che viene moltiplicato per se stesso è $1/a$, mentre il valore di b viene decrementato di un'unità ad ogni passo di ricorsione. In questo modo si hanno esattamente b passi ricorsivi prima di chiamare la funzione con parametro b uguale a zero, corrispondente al caso base in cui il risultato viene moltiplicato per il valore 1.

Nel caso generale si ha perciò la seguente situazione (ricordando che in questo caso il valore di b iniziale è positivo e viene decrementato ad ogni passo fino ad arrivare a zero):

$$\begin{aligned} \text{MiaFunz}(a,b) &= \\ (\text{passo } 1) &= 1/a \text{ MiaFunz}(a,b-1)= \\ (\text{passo } 2) &= 1/a^2 * \text{MiaFunz}(a,b-2)= \\ (\text{passo } 3) &= 1/a^3 * \text{MiaFunz}(a,b-3)=... \\ (\text{passo } b-2) &= 1/a^{b-2} * \text{MiaFunz}(a,b-(b+2))= \\ (\text{passo } b-1) &= 1/a^{b-1} * \text{MiaFunz}(a,b-(b+1))= \\ (\text{passo } b) &= 1/a^b * \text{MiaFunz}(a,b-(b))= \\ &= 1/a^b = a^{-b} \end{aligned}$$



Esercizio 2

2. I risultati stampati a video sono:

$r = 8$

$r = 4$

$r = 2$

$r = 1$

$r = 0.5$

$r = 0.25$

$r = 0.125$

Giustificazione

Ad ogni iterazione del ciclo presente nello script viene chiamata la funzione **MiaFunz** con parametri **x** e **y**. Tale funzione produce come risultato un numero che viene assegnato alla variabile **r**. La mancanza di un punto e virgola nell'istruzione di assegnamento fa in modo che MATLAB stampi il valore assegnato a **r** in quell'iterazione del ciclo.

Il ciclo viene iterato per i valori di **y** che variano da -3 a 3 con incrementi di 1. La variabile **x** invece assume un valore costante assegnato prima del ciclo pari a 2. Per cui si hanno 7 invocazioni della funzione con le seguenti coppie di parametri **(x,y)**: (2,-3), (2,-2), (2,-1), (2,0), (2,1), (2,2), (2,3). I valori stampati a video si ottengono quindi ricordando dal punto 1 dell'esercizio che la funzione calcola x^{-y} .



Esercizio 3

Si conviene di rappresentare un'immagine mediante una matrice rettangolare di numeri, corrispondenti al colore dei suoi pixel (punti luminosi che compongono la figura). Si vuole progettare una funzione MATLAB/Octave di nome `combinaImmagini` che, ricevendo come parametri due matrici $f1$ ed $f2$ rappresentanti due immagini e due valori numerici C ed S , con $C < S$, rappresentanti due diversi colori, produce come risultato una terza figura $f3$, ottenuta da $f1$ ed $f2$ secondo la seguente regola.

Nelle posizioni (r, c) in cui $f1(r, c) < C$ ed $f2(r, c) > C$ si ha $f3(r, c) = f2(r, c) - f1(r, c)$ nelle posizioni (r, c) in cui $f1(r, c) > S$ ed $f2(r, c) < S$ si ha $f3(r, c) = f1(r, c) - f2(r, c)$ nelle posizioni rimanenti si ha $f3(r, c) = f1(r, c)$.

- Codificare la funzione `combinaImmagini`.
- Scrivere uno script che acquisisce le due matrici di partenza rispettivamente dai file `file1.mat` e `file2.mat` (si supponga che le due variabili contenenti le matrici al momento del salvataggio si chiamino `matr1` e `matr2`), richiama la funzione `combinaImmagini` e salva sul file `file3.mat` la matrice risultante.



Esercizio 3

Soluzione

Versione MATLAB/Octave

```
%funzione
function [f3]=combinaImmagini(f1, f2, C, S)
f3=f1;
f3(f1<C & f2>C)=f2(f1<C & f2>C)-f1(f1<C & f2>C);
f3(f1>S & f2<S)=f1(f1>S & f2<S)-f2(f1>S & f2<S);

%script
load file1.mat matr1;
load file2.mat matr2;
C= input('inserisci il valore per C: ');
S= input('inserisci il valore per S: ');
matr3 = combinaImmagini(matr1, matr2, C, S);
```



Esercizio 4

Dopo una gara automobilistica si ha come risultato una tabella le cui colonne rappresentano gli N partecipanti (numerati da 1 a N) e le righe gli M giri di pista effettuati. Il valore di ogni generica cella (i,j) della tabella rappresenta il tempo impiegato dal partecipante j per percorrere il giro i .

Supponendo che tale tabella sia stata importata in MATLAB come matrice A , si scrivano le istruzioni per:

- 1) calcolare il tempo totale medio che è stato impiegato dai partecipanti per completare la gara;
- 2) determinare il vincitore della gara (cioè il numero del partecipante il cui tempo di percorrenza totale è minore di quello degli altri partecipanti);
- 3) tracciare un grafico in cui l'asse delle x rappresenta i giri compiuti dal vincitore e l'asse delle y il tempo necessario per percorrere i giri.



Esercizio 4

Soluzione punto 1

```
tempomedio=mean(sum(A))
```

Soluzione punto 2

```
primo=find(sum(A)==min(sum(A)))
```

Soluzione punto 3

```
x=1:M;  
y=A(vincitore,:);  
plot(x,y)  
hold on  
plot(x,y,'*')
```

**Tutte il materiale sarà
disponibile sul mio sito
internet!**

www.alessandronacci.it

See You Next Time!



**Tutte il materiale sarà
disponibile sul mio sito
internet!**

www.alessandronacci.it

See You Next Time!

