

Esercizio 2 (20/7/2009)

Una matrice Matlab contiene numeri interi. Si vuole progettare una funzione che ricevendo la matrice e un array di numeri interi che rappresenta una sequenza, cerchi tale sequenza all'interno della matrice. La sequenza può essere disposta, nella matrice, verticalmente dall'alto verso il basso od orizzontalmente, da sinistra verso destra. La funzione deve avere la seguente intestazione:

```
function [riga, col, dir] = cercaInizioSequenza(matrice, seq)
```

se la sequenza è presente nella matrice allora *riga* e *col* indicano gli indici di riga e di colonna del suo primo elemento, mentre *dir* viene posto uguale al carattere 'v' se la sequenza è disposta verticalmente, 'o' se orizzontalmente (se la sequenza è presente ripetuta in più posizioni, i valori restituiti possono essere quelli relativi a una qualsiasi delle ripetizioni); se la sequenza non è presente, *riga* e *col* valgono entrambi 0 e *dir* vale 'n'.

1. Per codificare la funzione in questione, si sviluppano prima le due seguenti funzioni ausiliarie
function [pres] = verificaSeqOrizzontaleDaPosizione (matrice, seq, riga, col)
function [pres] = verificaSeqVerticaleDaPosizione (matrice, seq, riga, col)
che ricercano la sequenza nella matrice a partire da una posizione d'inizio precisa, in direzione orizzontale o verticale: *riga* e *col* sono il punto di inizio; il risultato *pres* vale 1 se la sequenza è presente, 0 altrimenti.
2. Successivamente si utilizzano queste due funzioni per codificare le due seguenti
function [riga, col] cercaInizioSeqOrizzontale = (matrice, seq)
function [riga, col] cercaInizioSeqVerticale = (matrice, seq)
che ricercano la sequenza in tutta la matrice con disposizione orizzontale e verticale, restituendo in *riga* e *col* le coordinate del punto d'inizio, se la sequenza viene trovata, o il valore 0 altrimenti.
3. Infine si codifica la funzione *cercaInizioSequenza* facendo uso delle due precedenti.

Soluzione

```
function [pres] = verificaSeqOrizzontaleDaPosizione (matrice, seq,  
riga, col)  
len=length(seq);  
pr=matrice(riga,col:col+len-1);  
pres=all(pr==seq);
```

```
function [pres] = verificaSeqVerticaleDaPosizione (matrice, seq,  
riga, col)  
len=length(seq);  
pr=matrice(riga:riga+len-1,col);  
pres=all(pr==seq');
```

```
function [riga, col] = cercaInizioSeqOrizzontale (matrice, seq)  
len = length(seq);  
[R,C] = size(matrice);  
for r = 1 : R  
    for c = 1: C-len+1  
        pres = verificaSeqOrizzontaleDaPosizione(matrice, seq, r, c);  
        if pres  
            riga=r;  
            col=c;  
            return;  
        end;  
    end;  
end;  
riga=0;  
col=0;
```

```

function [riga, col] = cercaInizioSeqVerticale (matrice, seq)
len = length(seq);
[R,C] = size(matrice);
for c = 1 : C
    for r = 1: R-len+1
        pres = verificaSeqVerticaleDaPosizione(matrice, seq, r, c);
        if pres
            riga=r;
            col=c;
            return;
        end;
    end;
end;
riga=0;
col=0;

function [riga, col, dir] = cercaInizioSequenza(matrice, seq)
[ro, co] = cercaInizioSeqOrizzontale(matrice, seq);
if ro ~= 0
    riga = ro;
    col = co;
    dir = 'o';
    return;
end;
[rv, cv] = cercaInizioSeqVerticale(matrice, seq);
if rv ~= 0
    riga = rv;
    col = cv;
    dir = 'v';
    return;
end;
riga=0;
col=0;
dir='n';

```

Esercizio 2 (17/2/2009)

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 $f3(r, c) = f1(r, c)$.

- a) Codificare la funzione `combinaImmagini`.
- b) 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.

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 1 (25/1/2010)

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 e' 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.

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,'*')
```

Esercizio 2 (21/7/2010)

Si implementi una funzione MATLAB `incdec(s,t)` che riceve in ingresso un array numerico s ed un numero intero positivo t .

La funzione deve restituire:

1. il numero di incrementi a t passi in s
2. il numero di decrementi a t passi in s
3. il valore medio degli incrementi a t passi in s
4. il valore medio dei decrementi a t passi in s

Dove un incremento [decremento] a t passi in s è definito come una differenza positiva [negativa] tra il valore assunto da s in posizione $i+t$ e il valore assunto da s in posizione i .

Esempio. Sia $s=[4 \ 6 \ 9 \ 3 \ 9 \ 2]$ e $t=2$, abbiamo in s un solo incremento a 2 passi ($s(1)=4 \rightarrow s(3)=9$), due decrementi a 2 passi ($s(2)=6 \rightarrow s(4)=3$ ed $s(4)=3 \rightarrow s(6)=2$). Il valore medio degli incrementi è quindi 5, mentre quello dei decrementi è pari a -2 (cioè la media fra -3 e -1).

Soluzione

```
function [ninc,ndec,avginc,avgdec]=incdec(s,t)

diff = s(1+t:end) - s(1:end-t);
ninc=sum(diff>0);
avginc=mean(diff(diff>0));
ndec=sum(diff<0);
avgdec=mean(diff(diff<0));
```