

Esame di Programmazione Informatica

28 giugno 2024

Esercizio 1 (20/30)

Un'immagine in scala di grigi può essere approssimata graficamente da una matrice di caratteri ASCII. Tale rappresentazione si può ottenere attraverso la scrittura di una funzione MATLAB che prende in ingresso:

- una matrice $\mathbf{M} = [m_{ij}]$ di valori numerici m_{ij} compresi tra 0 ed 1,
- un vettore $\mathbf{w} = [w_1, \dots, w_n]$ di n caratteri w_i ,

e restituisce come argomento in uscita una matrice $\mathbf{C} = [c_{ij}]$ di caratteri, delle stesse dimensioni di $\mathbf{M}$, definita da $c_{ij} = w_k$. L'indice k di tale carattere w_k nel vettore $\mathbf{w}$ va scelto in proporzione al corrispondente valore di m_{ij} :

$$k = \begin{cases} 1 & \text{se } m_{ij} = 0 \\ n & \text{se } m_{ij} = 1 \\ \text{interpolazione lineare arrotondata all'intero più vicino} & \text{se } 0 < m_{ij} < 1 \end{cases}$$

Ad esempio, nel caso $n = 3$ (vettore $\mathbf{w}$ lungo 3 caratteri), si avrà $k = 1$ per $m_{ij} < \frac{1}{4}$, $k = 2$ per $\frac{1}{4} \leq m_{ij} < \frac{3}{4}$, $k = 3$ per $m_{ij} \geq \frac{3}{4}$. Ricordiamoci che una matrice di caratteri può essere ottenuta/inizializzata mediante la conversione di tipo `char(A)`, con $\mathbf{A}$ matrice dei corrispondenti valori interi nella codifica ASCII.

Implementare quindi la precedente funzione; mostrare poi come utilizzarla impiegando come matrice $\mathbf{M}$ quella fornita direttamente dalla seguente funzione che restituisce una matrice $m \times n$ dei valori numerici (compresi tra 0 ed 1) di luminosità dei pixel di un'immagine del pianeta Saturno:

`matriceImmagine.m`

```
function M = matriceImmagine(m, n)
    I = imread('saturn.png') ;
    I = imresize(I, [m, n]) ;
    M = double(I(:,:,1)) + double(I(:,:,2)) + double(I(:,:,3)) ;
    M = M - min( M(:) ) ;
    M = M / max( M(:) ) ;
end
```

nel caso di immagine con $m=20$ righe e $n=40$ colonne, e nel caso che il vettore di caratteri $\mathbf{w}$ sia definito da $w_1 = 'X'$, $w_2 = '*'$, $w_3 = '.'$, $w_4 = ' '$ (spazio). Come appare visivamente la matrice $\mathbf{C}$? Cosa succede visivamente a tale matrice invertendo l'ordine dei caratteri nel vettore $\mathbf{w}$?

Soluzione: ricaviamo la lunghezza n dal vettore **w** mediante la solita funzione **length()** ed implementiamo l'interpolazione lineare del tipo $ax + b \rightarrow (n-1)*M+1$ arrotondando all'intero più vicino mediante la funzione **round()**. La costruzione della matrice **C** può essere ottenuta direttamente mediante indicizzazione matriciale accedendo direttamente al vettore **w** con indice matriciale K:

```
matriceCaratteri.m
function C = matriceCaratteri(M, w)
    n = length(w) ;
    K = round( (n-1)*M+1 ) ;
    C = w(K) ;
end
```

Alternativamente si può operare mediante doppio ciclo **for**: inizializziamo la matrice di caratteri **C** con le dimensioni corrette per poi assegnare uno ad uno gli elementi nel doppio ciclo **for**:

```
matriceCaratteri.M
function C = matriceCaratteri(M, w)
    n = length(w) ;
    C = char(0*M) ;
    for i = 1:size(M,1)
        for j = 1:size(M,2)
            k = round( (n-1)*M(i,j)+1 ) ;
            C(i,j) = w(k) ;
        end
    end
end
```

Utilizzo nel caso considerato:

```
w = 'X*.' ;  
M = matriceImmagine(20, 40) ;  
C = matriceCaratteri(M, w)
```

ottenendo la seguente approssimazione grafica:

[illegible]

mentre l'inversione dei caratteri di $\mathbf{w}$ ($\mathbf{w} = \mathbf{w}^*$) determina l'ottenimento dell'immagine negativa.

Esercizio 2 (13/30)

La seguente potenza di binomio si può esprimere come segue:

$$(x+y)^n = \sum_{i=0}^n a_i x^i y^{n-i}$$

e può essere rappresentata in MATLAB dal vettore $\mathbf{a} = [a_0, \dots, a_n]$ dei coefficienti dell'espansione algebrica.

Il seguente prodotto è ancora una potenza di binomio:

$$(x+y)(x+y)^n = (x+y) \sum_{i=0}^n a_i x^i y^{n-i} = (x+y)^{n+1} = \sum_{i=0}^{n+1} b_i x^i y^{n+1-i}$$

che può quindi essere ancora rappresentato in MATLAB dal vettore $\mathbf{b} = [b_0, \dots, b_{n+1}]$ dei coefficienti dell'espansione algebrica. Tali coefficienti sono legati a quelli della precedente espansione dalla seguente relazione:

$$\mathbf{b} = [a_0, a_0 + a_1, a_1 + a_2, a_2 + a_3, \dots, a_{n-1} + a_n, a_n]$$

Scrivere una funzione MATLAB che prenda come argomento in ingresso il vettore $\mathbf{a}$, relativo a $(x+y)^n$, e restituisca come argomento in uscita il vettore $\mathbf{b}$, relativo a $(x+y)^{n+1}$. Si mostri poi come utilizzare questa funzione per calcolare, mediante applicazioni successive, i coefficienti del binomio $(x+y)^4$ partendo dai coefficienti di $(x+y)^1 = x+y$. Confrontare i coefficienti trovati con quelli binomiali forniti da MATLAB, ossia `nchoosek(4,0)`, `nchoosek(4,1)`, ..., `nchoosek(4,4)`.

Soluzione: La formula riportata (triangolo di Tartaglia) si può implementare in maniera immediata mediante operazioni vettoriali in quanto sarà sufficiente aggiungere uno 0 in testa ed uno 0 in coda al vettore $\mathbf{a}$ (supponendo vettori riga) per ottenere la somma voluta dei coefficienti intermedi:

`coefficientiProdotto.m`

```
function b = coefficientiProdotto(a)
    b = [a 0] + [0 a] ;
end
```

oppure sommando i coefficienti intermedi ed assegnando esplicitamente il primo e l'ultimo, facendo attenzione agli indici (ricordiamo che in MATLAB gli indici partono sempre da 1):

`coefficientiProdotto.m`

```
function b = coefficientiProdotto(a)
    b = [a(1) a(1:end-1)+a(2:end) a(end)] ;
end
```

Utilizzo nel caso di $n = 4$ partendo da $n = 1$:

```
coeff = [1 1] ;
for n = 2:4
    coeff = coefficientiProdotto(coeff)
end
```

ottenendo i coefficienti $[1,4,6,4,1]$ corrispondenti ai coefficienti binomiali forniti dalla funzione `nchoosek(4,0), nchoosek(4,1),..., nchoosek(4,4)`.