

RELAZIONE TERZA ESERCITAZIONE

- 1) REALIZZARE UNA M-FUNCTION CHE PRESO IN INPUT UN VETTORE, NE RESTITUISCA LA NORMA 1 E LA NORMA INFINITO

CODICE MATLAB , RISULTATI E ANALISI:

Il codice Matlab della funzione è il seguente:

```
function [ vuno,vinf ] = norma_uno( vet )  
% Restituisce la norma uno e la norma infinito dato un vettore in  
%input  
vuno = sum(abs(vet));  
vinf = max(abs(vet));  
end
```

Esempio di utilizzo:

```
v =      1      4      7     -9      2
```

```
[a,b] = norma_vett(v)
```

```
a =      23
```

```
b =       9
```

"a" è la norma 1, "b" è la norma infinito.

Per calcolare la norma 1 di un vettore si fa la somma del valore assoluto di tutti gli elementi del vettore, mentre la norma infinito è l'elemento del vettore che ha valore assoluto maggiore.

I risultati sono corretti, infatti:

norma 1 = $|1| + |4| + |7| + |-9| + |2| = 23$

norma infinito = $\max(|1|, |4|, |7|, |-9|, |2|) = 9$

2) REALIZZARE UNA M-FUNCTION CHE PRESA IN INPUT UNA MATRICE, NE RESTITUISCA LA NORMA 1 E LA NORMA INFINITO

CODICE MATLAB , RISULTATI E ANALISI:

Il codice Matlab della funzione è il seguente:

```
function [ muno,minf ] = norma_mat( mat )
% Restituisce la norma uno dato un vettore in input
muno = max(sum(abs(mat))); %somma per colonne
minf = max(sum(abs(mat),2)); %somma per righe
end
```

Esempio di utilizzo:

```
m =      5      6      7      2     -3
      3     -4      1     -9      2
```

```
[a,b] = norma_mat(m)
```

```
a =      11
```

```
b =      23
```

"a" è la norma 1, "b" è la norma infinito.

Per calcolare la norma 1 di una matrice si fa la somma dei valori assoluti di tutti gli elementi delle colonne e poi si prende il valore massimo, mentre la norma infinito si calcola facendo la somma dei valori assoluti delle righe e si prende poi il valore massimo.

I risultati sono corretti, infatti:

norma 1 = 11, cioè:

$$\max((|5| + |3|), (|6| + |-4|), (|7| + |1|), (|2| + |-9|), (|-3| + |2|))$$

norma infinito = 23, cioè:

$$\max((|5| + |6| + |7| + |2| + |-3|), (|3| + |-4| + |1| + |-9| + |2|))$$

- 3) VERIFICATE SU MATRICI A VOSTRA SCELTA LA VALIDITA' DELLA SEGUENTE RELAZIONE DI EQUIVALENZA TRA NORME DI MATRICI QUADRATE DI ORDINE N:

$$\frac{1}{n} \|A\|_{\infty} \leq \|A\|_1 \leq \sqrt{n} \|A\|_{\infty}$$

CODICE MATLAB , RISULTATI E ANALISI:

Il codice Matlab crea a random 10 matrici con elementi compresi fra 0 e 99 di ordine n (n è un parametro), poi per ogni matrice calcola la norma 1 e la norma infinito, controllando poi la relazione di equivalenza.

Questa funzione in output restituisce le 10 matrici generate e la relazione di ogni matrice:

```
function [ str1,str2,str3,str4,str5,str6,str7,str8,str9,str10 ] =  
equivalenza(n)
```

```
%crea 10 matrici random nxn
```

```
m1 = round(rand(n).*100);  
m2 = round(rand(n).*100);  
m3 = round(rand(n).*100);  
m4 = round(rand(n).*100);  
m5 = round(rand(n).*100);  
m6 = round(rand(n).*100);  
m7 = round(rand(n).*100);  
m8 = round(rand(n).*100);  
m9 = round(rand(n).*100);  
m10 = round(rand(n).*100);
```

```
%per ogni matrice calcola la norma 1 e la norma infinito, infine  
controlla
```

```
%la relazione di equivalenza
```

```
%a è la norma 1, b è la norma infinito
```

```
[a1,b1] = norma_mat(m1);  
[a2,b2] = norma_mat(m2);  
[a3,b3] = norma_mat(m3);  
[a4,b4] = norma_mat(m4);  
[a5,b5] = norma_mat(m5);  
[a6,b6] = norma_mat(m6);  
[a7,b7] = norma_mat(m7);  
[a8,b8] = norma_mat(m8);  
[a9,b9] = norma_mat(m9);  
[a10,b10] = norma_mat(m10);
```

```

%stampa tutte le matrici generate
m1
m2
m3
m4
m5
m6
m7
m8
m9
m10

%confronta la relazione di equivalenza per ogni matrice
str1 = strcat('1/n * norma inf = ',num2str(b1*(1/n)),' <= norma 1
= ',num2str(a1),' <= n^(1/2)*norma inf = ',num2str(sqrt(n)*b1));
str2 = strcat('1/n * norma inf = ',num2str(b2*(1/n)),' <= norma 1
= ',num2str(a2),' <= n^(1/2)*norma inf = ',num2str(sqrt(n)*b2));
str3 = strcat('1/n * norma inf = ',num2str(b3*(1/n)),' <= norma 1
= ',num2str(a3),' <= n^(1/2)*norma inf = ',num2str(sqrt(n)*b3));
str4 = strcat('1/n * norma inf = ',num2str(b4*(1/n)),' <= norma 1
= ',num2str(a4),' <= n^(1/2)*norma inf = ',num2str(sqrt(n)*b4));
str5 = strcat('1/n * norma inf = ',num2str(b5*(1/n)),' <= norma 1
= ',num2str(a5),' <= n^(1/2)*norma inf = ',num2str(sqrt(n)*b5));
str6 = strcat('1/n * norma inf = ',num2str(b6*(1/n)),' <= norma 1
= ',num2str(a6),' <= n^(1/2)*norma inf = ',num2str(sqrt(n)*b6));
str7 = strcat('1/n * norma inf = ',num2str(b7*(1/n)),' <= norma 1
= ',num2str(a7),' <= n^(1/2)*norma inf = ',num2str(sqrt(n)*b7));
str8 = strcat('1/n * norma inf = ',num2str(b8*(1/n)),' <= norma 1
= ',num2str(a8),' <= n^(1/2)*norma inf = ',num2str(sqrt(n)*b8));
str9 = strcat('1/n * norma inf = ',num2str(b9*(1/n)),' <= norma 1
= ',num2str(a9),' <= n^(1/2)*norma inf = ',num2str(sqrt(n)*b9));
str10 = strcat('1/n * norma inf = ',num2str(b10*(1/n)),' <= norma
1 = ',num2str(a10),' <= n^(1/2)*norma inf =
',num2str(sqrt(n)*b10));

end

```

Un esempio di utilizzo è:

```
[s1,s2,s3,s4,s5,s6,s7,s8,s9,s10] = equivalenza(5)
```

L'output della funzione è:

m1 =

81	10	16	14	66
91	28	97	42	4
13	55	96	92	85
91	96	49	79	93
63	96	80	96	68

m2 =

76	71	82	44	49
74	3	69	38	45
39	28	32	77	65
66	5	95	80	71
17	10	3	19	75

m3 =

28	50	75	96	84
68	96	26	55	25
66	34	51	14	81
16	59	70	15	24
12	22	89	26	93

m4 =

35	35	29	8	13
20	83	76	5	57
25	59	75	53	47
62	55	38	78	1
47	92	57	93	34

m5 =

16	60	45	83	11
79	26	8	54	96
31	65	23	100	0
53	69	91	8	77
17	75	15	44	82

m6 =

87	43	14	85	8
8	91	87	62	24
40	18	58	35	12
26	26	55	51	18
80	15	14	40	24

m7 =

42	49	78	13	23
5	34	39	94	35
90	90	24	96	82
94	37	40	58	2
49	11	10	6	4

m8 =

17	55	18	93	31
65	30	37	78	51
73	74	63	49	51
65	19	78	44	82
45	69	8	45	79

m9 =

64	94	21	19	31
38	88	30	23	92
81	55	47	17	43
53	62	23	23	18
35	59	84	44	90

m10 =

98	59	12	9	73
44	26	30	26	49
11	60	32	80	58
26	71	42	3	24
41	22	51	93	46

s1 =

$1/n * \text{norma inf} = 81.6 \leq \text{norma 1} = 339 \leq n^{(1/2)} * \text{norma inf} =$

912.3157

s2 =

$1/n * \text{norma inf} = 64.4 \leq \text{norma 1} = 305 \leq n^{(1/2)} * \text{norma inf} =$

720.0139

s3 =

$1/n * \text{norma inf} = 66.6 \leq \text{norma 1} = 311 \leq n^{(1/2)} * \text{norma inf} =$

744.6106

s4 =

$1/n * \text{norma inf} = 64.6 \leq \text{norma 1} = 324 \leq n^{(1/2)} * \text{norma inf} =$

722.25

s5 =

$$\frac{1}{n} * \text{norma inf} = 59.6 \leq \text{norma 1} = 295 \leq n^{(1/2)} * \text{norma inf} = 666.3483$$

s6 =

$$\frac{1}{n} * \text{norma inf} = 54.4 \leq \text{norma 1} = 273 \leq n^{(1/2)} * \text{norma inf} = 608.2105$$

s7 =

$$\frac{1}{n} * \text{norma inf} = 76.4 \leq \text{norma 1} = 280 \leq n^{(1/2)} * \text{norma inf} = 854.178$$

s8 =

$$\frac{1}{n} * \text{norma inf} = 62 \leq \text{norma 1} = 309 \leq n^{(1/2)} * \text{norma inf} = 693.1811$$

s9 =

$$\frac{1}{n} * \text{norma inf} = 62.4 \leq \text{norma 1} = 358 \leq n^{(1/2)} * \text{norma inf} = 697.6532$$

s10 =

$$\frac{1}{n} * \text{norma inf} = 50.6 \leq \text{norma 1} = 250 \leq n^{(1/2)} * \text{norma inf} = 565.7252$$

Come si vede dai risultati, la relazione di equivalenza è sempre verificata, quindi la relazione è valida.

4) COM' E' DEFINITA LA NORMA 1 DI UN VETTORE ?

$$\|x\|_1 = \sum_{i=1}^n |x_i|$$

5) COM' E' DEFINITA LA NORMA INFINITO DI UN VETTORE ?

$$\|x\|_\infty = \max_i |x_i|$$

6) COM' E' DEFINITA LA NORMA 1 DI UNA MATRICE ?

$$\|A\|_1 = \max_j \sum_{i=1}^n |a_{ij}|$$

7) COM' E' DEFINITA LA NORMA INFINITO DI UNA MATRICE ?

$$\|A\|_\infty = \max_i \sum_{j=1}^n |a_{ij}|$$

8) CHE RELAZIONE C' E' FRA LA NORMA 1 DI UNA MATRICE E LA NORMA INFINITO DELLA SUA TRASPOSTA ?

La norma 1 di una matrice è uguale alla norma infinito della sua trasposta, infatti la norma 1 di una matrice si calcola facendo per ogni colonna la somma dei valori assoluti degli elementi, poi si prende il valore massimo. Se la matrice viene trasposta, le sue colonne diventano le sue righe, quindi la norma infinito, che si calcola facendo per ogni riga la somma dei valori assoluti degli elementi e prendendo poi il valore massimo, è uguale alla norma 1 della matrice originale.

9) REALIZZARE UNA M-FUNCTION CHE EFFETTUA IL PRODOTTO DI UNA MATRICE SPARSA MEMORIZZATA IN FORMATO COMPRESSO CRS ED UN VETTORE

CODICE MATLAB , RISULTATI E ANALISI:

Il codice Matlab della funzione è il seguente:

```
function [ prod ] = sparsa( r, c, i , vet )
% vettore risultato, bisogna controllare col vettore i se bisogna
fare
% eventuali somme tra gli elementi del vettore tmp
tmp = r.*vet(c);
count = 1;
% vettore che contiene il risultato finale
prod = zeros(1,length(i)-1);
% scorre il vettore i
for k=1:1:length(i)-1
    stmp = 0;
    % controlla se la differenza dei 2 elementi consecutivi è > 1
    if i(k+1) - i(k) > 1
        % somma tanti elementi quant'è la differenza
        for j=count:1:count + i(k+1)-i(k) - 1
            stmp = stmp + tmp(j);
        end
        count = count + i(k+1)-i(k);
        prod(k) = stmp;
    else
        prod(k) = tmp(count);
        count = count + 1;
    end
end
end
```

Esempio di utilizzo:

Viene considerata questa matrice sparsa:

```
A =  8      0      0      0      0      0
      0      5      0      0      0      0
      0      0      0      3      5      0
      9      0      1      0      0      0
      0      3      0      0      0      2
      0      0      0      0      7      0
      0      0      9      0      0      3
```

Viene creato il vettore in colonna R, che contiene tutti gli elementi della matrice sparsa non nulli (quindi ha dimensione uguale al numero di elementi non nulli):

```
R = [8;5;3;5;9;1;3;2;7;9;3]
```

```
R =  8
      5
      3
      5
      9
      1
      3
      2
      7
      9
      3
```

Viene creato il vettore C, che contiene gli indici di colonna degli elementi di R (quindi ha la stessa dimensione di R):

```
C = [1,2,4,5,1,3,2,6,5,3,6]
```

```
C =  1      2      4      5      1      3      2      6      5      3      6
```

Viene creato il vettore I, che ha dimensione uguale al numero delle righe della matrice + 1. Per ogni riga considera il primo elemento non nullo, nel vettore è poi salvato il numero degli elementi non nulli della matrice + 1 prima di quell'elemento. Nel vettore è salvato anche il numero degli elementi non nulli + 1 della matrice rispetto all'ultimo elemento della sua ultima riga (quindi può anche non essere il primo della riga)

```
I = [1,2,3,5,7,9,10,12]
```

```
I =  1      2      3      5      7      9      10      12
```

Viene creato il vettore X, che è il vettore che viene moltiplicato per la matrice sparsa A:

```
X = [1;0;2;3;1;1]
```

```
X = 1
     0
     2
     3
     1
     1
```

Il risultato è:

```
sparsa(R,C,I,X)
```

```
ans =      8      0     14     11      2      7     21
```

che è corretto, infatti se viene fatto il prodotto di $A * X$ il risultato è:

```
A*X
```

```
ans =      8
          0
          14
          11
           2
           7
          21
```

FUNZIONAMENTO:

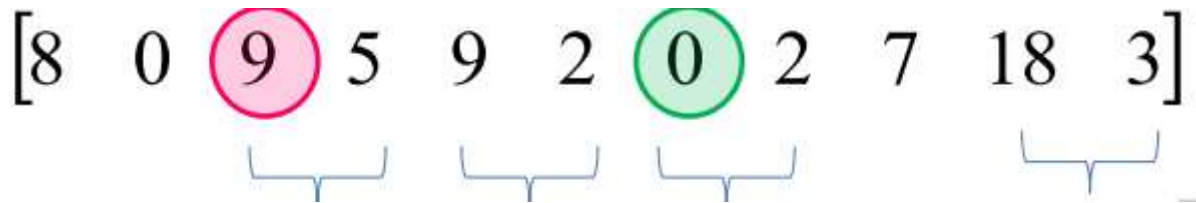
Per prima cosa viene fatto il prodotto fra gli elementi di R e gli elementi di X di indice C, ad esempio:

```
R(3,1) = 3      C(1,3) = 4      X(4,1) = 3
```

Viene fatto $R(3,1) * X(4,1) = 3 * 3 = 9$, e il risultato viene salvato nel vettore temporaneo (tmp, ha la stessa dimensione di R) all'indice 3.

Il vettore temporaneo ora contiene alcuni elementi che devono essere sommati tra di loro, cioè i prodotti degli elementi che erano nella stessa riga della matrice A.

In particolare, devono essere fatte queste somme:



E bisogna quindi creare un nuovo vettore (prod) che contenga i nuovi risultati, che sono il prodotto fra la matrice A ed il vettore X.

Per sapere quali sono gli elementi da sommare, si scorre il vettore I e si guarda la differenza fra i 2 elementi consecutivi. Se la differenza è maggiore di 1, allora nella riga c'era più di un elemento, e gli elementi del vettore temporaneo devono essere salvati.

Esempio:

i primi elementi di I sono 1 2 3 5, partendo dal primo elemento si trova:

$2 - 1 = 1 \Rightarrow$ si copia nel vettore prod il primo elemento di R

$3 - 2 = 1 \Rightarrow$ si copia nel vettore prod il secondo elemento di R

$5 - 3 = 2 \Rightarrow$ si fa la somma fra il terzo ed il quarto elemento di R, poi si salva il risultato nel vettore prod.

Se la differenza era 3 si sommano i 3 elementi consecutivi di R.

A questo punto l'indice che scorre il vettore R viene spostato sul prossimo elemento da controllare, cioè il quinto.

Si continua così finché non si arriva al penultimo elemento del vettore I, poi il vettore prod è completato e si è trovato il risultato di $A * X$.