Fondamenti di Informatica, A.A. 2011-2012

11/02/2013

Esercizio 1

È dato il frammento di codice Matlab

```
n = 3;     v = zeros(n,1);
w = zeros(1,n);     a = zeros(n,n);
for i=1:n
  v(i) = 3;
  w(i) = n-i;
  for j=1:n
    a(i,j) = (i-1)*n+j;
  end
end
a = a+v*w;     disp(a);
```

Si chiede cosa viene visualizzato dall'interprete Matlab.

Soluzione

L'interprete Matlab stampa:

```
     7     5     3
    10     8     6
    13    11     9
```

Esercizio 2

Discutere delle convenzioni di passaggio dei parametri delle funzioni in Matlab.

Soluzione

Il passaggio dei parametri alle funzioni in Matlab avviene normalmente per valore: il valore della variabile del chiamante viene copiato nello spazio di memoria della funzione. Nel caso di passaggio di dati matriciali, l'operazione di copia comporta un costo che può diventare rilevante. Qualsiasi alterazione degli argomenti effettuata dalla funzione viene quindi ignorata dalla funzione chiamante, la quale viene influenzata esclusivamente attraverso i parametri di uscita.

Il linguaggio Matlab fornisce anche la possibilità di utilizzare un numero di argomenti variabile sia in ingresso che in uscita. Data una funzione che dichiari un certo numero di parametri di ingresso e di uscita, all'interno della funzione è possibile utilizzare le funzioni `nargin` e `nargout` per accedere all'effettivo numero di parametri che sono stati specificati nella chiamata, in modo da poter gestire degli opportuni valori di default ove necessario.

Esercizio 3

In una prova di esame di informatica viene richiesto di trovare, in un vettore contenente solo zeri ed uni, la dimensione della più corta sequenza di zeri. Uno studente propone il seguente codice

```
count = 0;    minl = length(v)+1;
for i=1:length(v)
    if (v(i)==1)
        if (count <minl)
            minl=count;
        end
        count=0;
    else
        count=count+1;
    end
end
if (v(length(v))==0)
    if (count < minl)
        minl=count;
    end
end
disp("La minima lunghezza è");
disp(minl);
```

Si chiede di trovare l'errore presente nel codice e di proporre una correzione adeguata.

Soluzione

Il problema del codice è che la terminazione di una sequenza di zeri viene riconosciuta dalla presenza di un uno. Se il primo elemento del vettore vale 1, allora il valore di `minl` viene immediatamente aggiornato a 0 indipendentemente dalla effettiva presenza nel vettore di sottosequenze di zeri. Una correzione possibile consiste nello scorrere tutti gli elementi di valore 1 in testa al vettore

```
count = 0;    minl = length(v)+1;
j=1;
while ((v(j)==1)&&(j<=length(v))
  j=j+1;
end
for i=j:length(v)
    if (v(i)==1)
        if (count <minl)
            minl=count;
        end
        count=0;
    else
        count=count+1;
    end
```

```
  end
  if (v(length(v))==0)
     if (count < minl)
        minl=count;
     end
  end
  disp('La minima lunghezza è');
  disp(minl);
```

In questo caso se il vettore contiene solo valori 1 il valore stampato è `length(v)+1` che può essere interpretato come assenza di zeri.

Esercizio 4

È data la seguente funzione Matlab

```
function [x]=simple(a,x,b,m,thr,imax)
  n=length(x);
  if ((size(a,1) ~= n)||(size(a,2) ~= n)||(size(m,1) ~= n)||(size(m,2) ~= n) )
     disp('size mismatch');
     return
  end
  r=b-a*x;
  i=0;
  while ((i<imax)&&(norm(r)>thr))
     x=x+m\r;
     r=b-a*x;
     i=i+1;
  end
end
```

Si stimi il numero di operazioni aritmetiche eseguite, assumendo come parametro il numero di iterazioni compiute nel ciclo `while`; si assuma inoltre che l'operatore `norm` su un vettore di lunghezza n abbia costo $2n$. Si supponga che `m` sia una matrice triangolare.

Soluzione

La soluzione di un sistema triangolare di dimensione n costa n^2. All'interno del ciclo `while` vengono eseguite una risoluzione di sistema, ed il calcolo di un residuo, che viene effettuato anche una volta prima della esecuzione del ciclo. Il calcolo del residuo comporta un prodotto matrice-vettore di costo $2n^2 - n$ e la somma ad un altro vettore di costo n, in definitiva il costo è $2n^2$. Infine la valutazione della norma del residuo viene effettuata per ogni iterazione del ciclo `while` più una volta quando viene riconosciuta l'uscita dal ciclo (trascuriamo in prima approssimazione il caso di uscita per `i>imax`). Denotando con k il numero di iterazioni effettivamente eseguito, abbiamo dunque

$$OP = (k+1) \cdot 2n^2 + k \cdot n^2 + (k+1)n \approx 3kn^2.$$

Esercizio 5

È dato il frammento di codice Matlab

```
i=1;
so=0;
no=0;
se=0;
ne=0;
while (i<=length(v))
    if (mod(i,2)==1)
        so = so + v(i)^2;
        no = no + 1;
    else
        se = se + (-v(i))^3;
        ne = ne + 1;
    end
    i = i + 1;
end
s=se-so
```

Riscrivere il codice facendo uso di uno o piu' cicli `for` e senza usare la istruzione `if`. Bonus: riscrivere usando solo operazioni su array.

Soluzione

La istruzione `if` serve a separare le iterazioni pari da quelle dispari; lo stesso risultato può essere ottenuto specificando un incremento nel ciclo `for`, come segue

```
i=1;
so=0;
no=0;
se=0;
ne=0;
for i=1:2:length(v)
    so = so + v(i)^2;
    no = no + 1;
end
for i=2:2:length(v)
    se = se + (-v(i))^3;
    ne = ne + 1;
end
s=se-so
```

ovvero, usando direttamente il linguaggio degli array,

```
s=sum((-v(2:2:length(v))).^3)-sum(v(1:2:length(v)).^2)
```