

Fondamenti di Informatica, A.A. 2011-2012

23/07/2012 — Fila A

Esercizio 1

È dato il codice matlab

```
n = 6;  
v = zeros(n,1);  
for i=1:n  
    v(i) = max(0,n/2-i)+ max(0,i-n/2);  
end  
disp(v);
```

Si chiede cosa viene visualizzato dall'interprete Matlab.

Soluzione

v è un vettore colonna che assumerà in una prima parte di valori discendenti dati da $\max(0, n/2-i)$ ed in una seconda ascendenti dati da $\max(0, i-n/2)$, e per la precisione:

```
2  
1  
0  
1  
2  
3
```

Esercizio 2

Descrivere le caratteristiche principali delle funzioni ricorsive.

Soluzione

Una funzione si dice ricorsiva quando richiama direttamente o indirettamente se stessa; pertanto potranno aversi diverse copie della funzione in esecuzione contemporaneamente. Ogni "istanza" di una funzione ricorsiva ha assegnato uno spazio di memoria (detto "stack frame") separato da quello di tutte le altre istanze. Nel caso particolare in cui la chiamata ricorsiva sia l'ultima istruzione eseguibile si parla di "tail recursion"; in questo caso la nuova istanza può riutilizzare lo spazio di memoria precedente, con un considerevole risparmio di tempo e memoria complessiva. Spesso le funzioni ricorsive hanno bisogno di una funzione di preparazione, detta "wrapper".

Esercizio 3

È data il seguente frammento Matlab

```
if (~ (a>b)) && (b~=c)  
    disp(1)  
else
```

```
disp(2)
end
```

Si chiede se viene stampato il valore 1 oppure 2 per i seguenti valori di ingresso:

1. $a = 2; b = 3; c = 2;$
2. $a = 3; b = 5; c = 5;$
3. $a = 5; b = 4; c = 3;$

Soluzione

1. Stampa 1;
2. Stampa 2;
3. Stampa 2.

Esercizio 4

È data la seguente funzione Matlab

```
function [x]=mystery(a,x,y)
k=size(x,2);
for i=1:k
a = a - x(:,i)*y(i,:);
end
end
```

Stimare il numero di operazioni aritmetiche ipotizzando che a sia una matrice $n \times n$, x sia una matrice $n \times k$ e y sia una matrice $k \times n$.

Soluzione

All'interno del ciclo su k viene eseguito un prodotto tra un vettore colonna ed un vettore riga; questa operazione produce una matrice quadrata $xy(i, j) = x(i) * y(j)$; il numero di operazioni è quindi n^2 . Questa matrice viene poi sottratta dalla matrice a , elemento per elemento, per ulteriori n^2 operazioni. Siccome il ciclo viene eseguito k volte il numero totale di operazioni sarà

$$OP = 2kn^2$$

La stessa conclusione si applica anche osservando che il codice è equivalente a

```
function [x]=mystery(a,x,y)
k=size(x,2);
a = a - x(:,1:k)*y(1:k,:);
end
```

Esercizio 5

È dato il frammento di codice Matlab

```
k=1;
m=0;
while (k<n)
    if (abs(v(k))>m)
        m=abs(v(k));
    end
    k=k+2;
end
```

Riscrivere il codice facendo uso del ciclo `for`. Bonus: riscrivere il codice facendo uso diretto delle operazioni su array.

Soluzione

Il codice calcola il massimo valore assoluto degli elementi di posto dispari dell'array v , fino ad $n - 1$; naturalmente se k arrivasse ad assumere un valore maggiore della dimensione di v si avrebbe un errore di indicizzazione. Il codice equivalente che usa `for` è quindi

```
m=0;
for k=1:2:n-1
    if (abs(v(k))>m)
        m=abs(v(k));
    end
end
```

ma lo stesso risultato poteva essere ottenuto con

```
m=max(abs(v(1:2:n-1)));
```

Fondamenti di Informatica, A.A. 2011-2012

09/07/2012 — Fila B

Esercizio 1

È dato il codice matlab

```
n = 6;  
v = zeros(n,1);  
for i=1:n  
    v(i) = min(n/2,i) - max(0,i-n/2);  
end  
disp(v);
```

Si chiede cosa viene visualizzato dall'interprete Matlab.

Soluzione

v è un vettore colonna che assumerà in una prima parte di valori ascendenti dati da $\min(n/2,i)$ ed in una seconda parte i valori ridiscenderanno per effetto di $\max(0,i-n/2)$, e per la precisione:

```
1  
2  
3  
2  
1  
0
```

Esercizio 2

Descrivere le caratteristiche principali delle funzioni Matlab, in particolare riguardo alla gestione della memoria.

Soluzione

Le `function` di Matlab realizzano delle trasformazioni di dati in maniera autocontenuta, e possono essere usate per strutturare appropriatamente dei codici complessi. In Matlab le funzioni hanno un proprio spazio di indirizzamento separato dallo spazio chiamante; le variabili contenute in questo spazio di indirizzamento sono visibili solo dall'interno della funzione stessa.

La funzione comunica con il resto dell'ambiente solo attraverso i suoi argomenti di ingresso e di uscita; sia in ingresso che in uscita degli argomenti viene effettuata una copia. Questo implica che le eventuali modifiche ad uno degli argomenti di ingresso rimangono confinate alla funzione stessa e non si riflettono sul valore dell'argomento nello spazio del programma chiamante. Questa caratteristica è molto comoda ma comporta anche un costo nel passaggio dei parametri che in alcuni casi può essere abbastanza pesante.

Esercizio 3

È data il seguente frammento Matlab

```
if (~ (a<=b)) || (b~=c)
    disp(1)
else
    disp(2)
end
```

Si chiede se viene stampato il valore 1 oppure 2 per i seguenti valori di ingresso:

1. $a = 2; b = 2; c = 2;$

2. $a = 3; b = 5; c = 5;$

3. $a = 5; b = 4; c = 3;$

Soluzione

1. Stampa 2;

2. Stampa 2;

3. Stampa 1.

Esercizio 4

È data la seguente funzione Matlab

```
function [x]=mystery(a,x,k)
for i=1:k
    y = a*x;
    alpha = x'*y;
    x = x + alpha*y;
end
```

Stimare il numero di operazioni aritmetiche ipotizzando che a sia una matrice $n \times n$ e x sia un vettore $n \times 1$.

Soluzione

Il ciclo viene eseguito per k volte; in ciascuna iterazione vengono effettuate le seguenti operazioni:

1. Un prodotto matrice-vettore $y=a*x$ che produce un nuovo vettore y di dimensione $n \times 1$; questo comporta $2n^2 - n$ operazioni;
2. Un prodotto scalare $x'*y$ che produce uno scalare con $2n - 1$ operazioni;
3. Una operazione AXPY (somma di un vettore con un multiplo di un altro vettore) che comporta 2 operazioni per ogni elemento del vettore, ossia $2n$.

In totale si avrà quindi:

$$OP = k \cdot (2n^2 + 3n - 1) = O(kn^2)$$

Esercizio 5

È dato il frammento di codice Matlab

```
k=n;  
m=inf;  
while (k>0)  
    if (v(k) <m)  
        m=v(k);  
    end  
    k=k-2;  
end
```

Riscrivere il codice facendo uso del ciclo `for`. Bonus: riscrivere il codice facendo uso diretto delle operazioni su array.

Soluzione

Il codice calcola il minimo valore degli elementi di posto da n a 1 a passo -2 , quindi di posto pari se n è pari, dispari altrimenti. Il valore del minimo viene inizializzato a `inf` che nella aritmetica usata da Matlab rappresenta ∞ . Naturalmente se n dovesse assumere un valore maggiore della dimensione di v si avrebbe un errore di indicizzazione. Il codice equivalente che usa `for` è

```
m=inf;  
for k=n:-2:1  
    if (v(k) <m)  
        m=v(k);  
    end  
end
```

ma lo stesso risultato poteva essere ottenuto con

```
m=min(v(n:-2:1));
```