Fondamenti di Informatica, A.A. 2013-2014 - FILA A

01/07/2014

Esercizio 1

È dato il frammento di codice Matlab

```
a = [16 32 64 128 256 512 1024 2048];
b = [8 16 24 32 40 48 56 64];

x = a(1:2:end);
y = b(end:-2:1);

res = [a(1:3:(end-1)) b(1:3:(end-1)) x(1:end-1)];

n = length(res)

disp(res(1:n-1))
```

Si chiede cosa viene visualizzato dall'interprete Matlab.

Soluzione ES.1

```
x = [16, 64, 256, 1024]
y = [64, 48, 32, 16]
res = [16, 128, 1024, 8, 32, 56, 16, 64, 256]
n = 9
disp(res(1:n-1)) = [16, 128, 1024, 8, 32, 56, 16, 64]
```

Esercizio 2

Descrivere il funzionamento dell'algoritmo di ordinamento Insertion Sort. Quale è la sua complessità computazionale?

Soluzione ES.2

Per la descrizione si vedano luci presentati a lezione. La complessità è $O(n^2)$ nel caso medio e peggiore.

Esercizio 3

Una matrice si dice a diagonale dominante per righe quando l'elemento sulla diagonale principale, in modulo, è maggiore della somma dei moduli degli altri elementi della riga. Per verificare la proprietà viene proposta la seguente funzione; si chiede di trovare l'errore presente nel codice e di proporre una correzione.

```
% Si supponga a una matrice
check = 1;
for i=1:size(a,1)
```

```
  for j=1:size(a,2)
    if(i==j)
      r = sum(abs(a),2);
      check = check * ((r(i)-a(i,j)) < abs(a(i,i)));
    end
  end
end
disp(check)
```

Soluzione ES.3

Il problema fondamentale sta nella assenza del modulodi `a(i,j)` nella sottrazione, per cui la correzione minima è

```
% Si supponga a una matrice
check = 1;
for i=1:size(a,1)
  for j=1:size(a,2)
    if(i==j)
      r = sum(abs(a),2);
      check = check * ((r(i)-abs(a(i,i))) < abs(a(i,i)));
    end
  end
end
disp(check)
```

Inoltre la struttura del codice è ridondante, si potrebbe utilmente semplificare come

```
% Si supponga a una matrice
check = 1;
for i=1:size(a,1)
    r = sum(abs(a(i,:)));
    check = check * ((r-abs(a(i,i))) < abs(a(i,i)));
end
disp(check)
```

Esercizio 4

È data la seguente funzione Matlab

```
function [x n]=simple(a,b)
  n=length(b);
  x = b;
  for i=1:n
    d = diag(a);
    d = d./2;
    a = a + sum(d);
    x = x + d;
 end
```

Si stimi il numero di operazioni aritmetiche eseguite, assumendo che a sia una matrice $n \times n$ e che b sia un array $n \times 1$.

0.1 Soluzione ES.4

```
function [x n]=simple(a,b)
  n=length(b);
  x = b;
  for i=1:n   % ciclo eseguito n volte
    d = diag(a);
    d = d./2;         % n operazioni aritmetiche
    a = a + sum(d); % n^2 +n operazioni aritmetiche
    x = x + d;        % n operazioni aritmetiche
 end
```

in totale vengono eseguite $n(n^2 + 3n)$ operazioni aritmetiche

Esercizio 5

È dato il frammento di codice Matlab

```
sel = input('Write␣FC␣for␣Fahrenheit␣to␣Celsius␣or␣CF␣viceversa\n','s');
temp = input('Insert␣the␣value\n');
switch sel
    case 'CF'
        temp = 1.8*temp + 32;
        str = [num2str(temp) '␣F␣degs'];
        disp(str);
    case 'FC'
        temp = (temp - 32)/1.8;
        str = [num2str(temp) '␣C␣degs'];
        disp(str);
    otherwise
        disp('Error');
end
```

Riscrivere il codice facendo uso di uno o più blocchi condizionali `if-then-else`.

0.2 Soluzione ES.5

Una possibile soluzione è la seguente:

```
sel = input('Write␣CF␣for␣Fahrenheit␣to␣Celsius␣or␣FC␣viceversa\n','s');
temp = input('Insert␣the␣value\n');
if (strcmp(sel,'CF'))
    temp = 1.8*temp + 32;
    str = [num2str(temp) '␣F␣degs'];
    disp(str);
elseif (strcmp(sel,'FC'))
```

```
    temp = (temp - 32)/1.8;
    str = [num2str(temp) ' C degs'];
    disp(str);
else
    disp('Error');
end
```

Fondamenti di Informatica, A.A. 2013-2014 - FILA B

01/07/2014

Esercizio 1

È dato il frammento di codice Matlab

```
v = [1 2 3 4];

for i=1:length(v)
  v = [sum(v) v];
  l = length(v);
  k = l-i;
  v(k) = v(end);
  v = v(2:end);
end

disp(v)
```

Si chiede cosa viene visualizzato dall'interprete Matlab.

Soluzione ES.1

Seguendo lo svolgimento della prima iterazione:

```
v =[    10     1     2     3     4]

l =  5
k =  4
v =[    10     1     2     4     4]

v =[    1     2     4     4]
```

Procedendo si continuano a sostituire i valori di `v` fino ad ottenere

v=[4 ,4 ,4 ,4]

Esercizio 2

Discutere come vengono rappresentate le stringhe in MATLAB/Octave, come è possibile manipolarle (concatenazione, troncamento, ecc...) e compararle. Avendo la definizione `str = '2013'` cosa restituisce l'istruzione MATLAB `str + 1`?

Soluzione esercizio 2

Applicando l'operatore somma alla stringa str essa viene trasformata in un vettore di lunghezza length(str) e contenente il codice ascii dei relavi caratteri in str. Dopo tale trasformazione viene eseguita l'operazione di somma tra un vettore ed uno scalare. Quindi il risultato sarà il vettore

[codiceAscii(2)+1, codiceAscii(0)+1, codiceAscii(1)+1, codiceAscii(3)+1]

che corrisponde alla stringa `'3124'`

Esercizio 3

Una parola si dice palindroma se è uguale sia letta da sinistra verso destra che da destra verso sinistra (es: 'rotor'). Viene propostala seguente funzione per verificare se una stringa sia palindroma.

```
function res = palindrome(str)

  len = length(str);

  res = 1;

  if(len==1)
      return;
  end

  res = res * palindrome(str(2:len-1));

  if(str(1) ~= str(len))
      res = 0;
      return
  end

end
```

Si chiede di trovare l'errore presente nel codice e di proporre una correzione adeguata. La funzione deve ritornare 1 se la stringa è palindroma o zero altrimenti.

Soluzione ES.3

Per evitare un loop infinito occorre che la base della ricorsione venga modificata in

```
...
  if(len<=1)
      return;
  end
...
```

Questa correzione è sufficiente a garantire la correttezza.

Il codice può inoltre essere modificato in modo da ottenere una *tail recursion*, e quindi migliorare l'efficienza, nel modo seguente

```
function res = palindrome(str)

  res = 1;
  len = length(str);

  if (len<=1)
      return;
  end

  if (str(1) ~= str(len))
      res = 0;
```

```
        return,
    end

    res = palindrome(str(2:len-1));
end
```

Esercizio 4

È data la seguente funzione Matlab

```
function [x]=mistery(m,y,k)
  for i=1:(g(k)./2)
      y = y + m^2;
  end
end
```

Si stimi il numero di operazioni aritmetiche eseguito dal codice, considerando le seguenti possibili scelte di $g(k)$

1. $g(k) = 4$;
2. $g(k) = 2 * k$;
3. $g(k) = 6 * k$.

Si assuma che m sia una matrice quadrata $n \times n$, che y sia uno scalare positivo.

0.3 Soluzione ES.4

L'operazione

y=y+m^2

richiede $2n^3 + n^2$ operazioni aritmetiche. Quindi

1. nel caso $g(k) = 4$; abbiamo $2(2n^3 + n^2)$
2. nel caso $g(k) = 2 * k$; abbiamo $k(2n^3 + n^2)$
3. nel caso $g(k) = 6 * k$; abbiamo $3k(2n^3 + n^2)$

Esercizio 5

Facendo uso del costrutto `switch` scrivere un frammento di codice MATLAB che determini se un giorno della settimana è feriale o festivo. Si assuma che i giorni della settimana siano codificati mediante un numero intero: 1 corrisponde al lunedì e 7 alla domanica. Si assume che i giorni festivi siano il sabato ed la domenica.

Soluzione ES.5

```
d=input('inserisci il numero del giorno della settimana:');
switch d
        case {1,2,3,4,5}
                disp('giorno feriale');
        case {6,7}
                disp('giorno festivo');
        otherwise
                disp('giorno errato');
end
```