

Fondamenti di Informatica, A.A. 2013–2014

02/02/2015

Esercizio 1

È dato il frammento di codice Matlab

```
a = 1;
n = 10;
i = a;
v = [0];
while(mod(i , n/2) != 0)
    v = [v a*i];
    l = length(v);
    a = l-a;
    i = i + a;
end
```

```
disp(v)
```

Si chiede cosa viene visualizzato dall'interprete Matlab (giustificando la risposta).

Soluzione

L'interprete Matlab stampa

```
0    1    2    8   12   27   36   64
```

Si noti che l'avanzamento di i non avviene per quantità costanti, quindi il ciclo *non* termina con $i=5$.

Esercizio 2

Descrivere il comportamento della funzione `linspace(a,b,c)` al variare dei suoi argomenti. Quali sono i suoi possibili usi?

Soluzione

La funzione `linspace(a,b,c)` crea un numero fissato c di valori, distribuiti linearmente, compresi fra i due limiti a e b (inclusi). Tale funzionalità è utile in tutte le situazioni in cui si debba generare una griglia di valori di dimensione predeterminata su cui calcolare una qualche funzione. Due esempi di tale situazione sono la creazione di grafici con la funzione `plot` e la integrazione numerica.

Esercizio 3

In una prova di esame di informatica viene richiesto di realizzare un programma che calcoli lo zero di una funzione f in un intervallo $[a, b]$.

Viene proposta la seguente funzione che usa il metodo della bisezione:

```

function [x e] = mybisect(f,a,b,n)
    % function [x e] = mybisect(f,a,b,n)
    % Does n iterations of the bisection method for a function f
    % Inputs: f — an inline function
    % a,b — left and right edges of the interval
    % n — the number of bisections to do.
    % Outputs: x — the estimated solution of  $f(x) = 0$ 
    % e — an upper bound on the error
    format long
    c = f(a); d = f(b);
    if c*d >= 0.0
        error('Function has same sign at both endpoints.')
    end
    for i = 1:n
        x = (a + b)/2;
        y = f(x);
        % disp([ x y]); % uncomment to debug
        if y == 0.0 % solved the equation exactly
            e = 0;
        end
        if c*y > 0
            b=x;
        else
            a=x;
        end
    end
    e = (b-a)/2;

```

Questa funzione contiene degli errori; si chiede di proporre una correzione adeguata. Bonus: modificare la funzione in modo che restituisca un risultato quando la stima dell'errore scende sotto una certa soglia specificata da utente.

Soluzione

Il codice proposto contiene i seguenti problemi:

1. Il controllo iniziale dà come errore anche il caso in cui la funzione sia nulla in uno degli estremi;
2. Nel caso in cui si raggiunga uno zero, non si esce immediatamente dal ciclo.

Una soluzione (comprensiva della uscita nel caso di errore minore di una soglia) può essere:

```

function [x e] = mybisect(f,a,b,n,tol)
    % function [x e] = mybisect(f,a,b,n)
    % Does n iterations of the bisection method for a function f
    % Inputs: f — an inline function
    % a,b — left and right edges of the interval

```

```

% n — the number of bisections to do.
% Outputs: x — the estimated solution of  $f(x) = 0$ 
% e — an upper bound on the error
if (nargin < 5)
    tol = 0
end
c = f(a); d = f(b);
if (c==0)
    x = a;
    e = 0;
    return;
end
if (d==0)
    x = b;
    e = 0;
    return;
end
if c*d > 0.0
    error('Function has same sign at both endpoints.')
end
for i = 1:n
    x = (a + b)/2;
    y = f(x);
    % disp([ x y]) % uncomment to debug
    e = (b-a)/2;
    if (e < tol)
        break;
    end
    if y == 0.0 % solved the equation exactly
        e = 0;
        break; % jumps out of the for loop
    end
    if c*y < 0
        b=x;
    else
        a=x;
    end
end
end

```

Esercizio 4

È data la seguente funzione Matlab

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

dove m e y sono matrici quadrate $n \times n$, ed x e k sono interi positivi. Si stimi il numero di operazioni aritmetiche eseguito dal codice, considerando le seguenti possibili scelte di $g(i)$

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

Soluzione

L'elevamento a potenza x di una matrice, nella sua versione più semplice, comporta una moltiplicazione matrice-matrice ripetuta $x - 1$ volte.¹ La moltiplicazione matrice-matrice $n \times n$ costa $2n^3$, quindi la potenza costa $2(x - 1)n^3$, se $x > 1$, altrimenti il calcolo di m^1 naturalmente non comporta alcuna operazione aritmetica. Ad ogni iterazione viene inoltre effettuata una somma di due matrici, che comporta ulteriori n^2 operazioni. In totale, per una singola iterazione del ciclo, si avranno $2(x - 1)n^3 + n^2$ operazioni aritmetiche.

Nel caso in cui $g(i) = 4$ verranno eseguite $g(k)/2 = 2$ iterazioni, e quindi si avranno in totale $2 * (2(x - 1)n^3 + n^2)$ operazioni, ovvero $O(xn^3)$.

Nel caso in cui $g(i) = 2 * i$ verranno eseguite $g(k)/2 = 2k/2 = k$ iterazioni, e quindi si avranno in totale $k * (2(x - 1)n^3 + n^2)$ operazioni, ovvero $O(kxn^3)$.

Nel caso in cui $g(i) = 6 * i$ verranno eseguite $g(k)/2 = 6k/2 = 3k$ iterazioni, e quindi si avranno in totale $3k * (2(x - 1)n^3 + n^2)$ operazioni, ovvero ancora $O(kxn^3)$.

Nel caso in cui $x = 1$, si avranno $2n^2$, kn^2 e $3kn^2$ operazioni, rispettivamente.

Esercizio 5

È dato il frammento di codice Matlab

```
ii = 1; jj = 0; kk = 0; hh = 0;

while (ii <= length(a))
    if mod(a(ii),2)==0
        jj = 12+ii;
    elseif mod(a(ii),3)==1
        jj = 11+ii;
    else
```

¹In realtà è possibile realizzare la potenza di matrici con un numero inferiore di prodotti, basandosi sulla rappresentazione binaria del numero x , ma ai fini di questo esercizio la moltiplicazione ripetuta è sufficiente

```

        kk = 11+ii;
        if mod(jj,12)==0
            hh = hh+1;
        end
    end
    ii = ii+1;
end

```

Riscrivere il codice facendo uso di uno o più cicli `for` ed utilizzando obbligatoriamente il costrutto `switch` al posto degli `if`.

Soluzione

Una possibile soluzione è la seguente:

```

ii = 1; jj = 0; kk = 0; hh = 0;
for ii = 1:length(a)
    switch mod(a(ii),2)
        case 0
            jj = 12+ii;
        otherwise
            switch mod(a(ii),3)
                case 1
                    jj = 11+ii;
                otherwise
                    kk = 11+ii;
                    switch mod(jj,12)
                        case 0
                            hh = hh+1;
                    end
            end
        end
    end
end
end

```