

# Hw1

劉弘祥 106022103 2018/11/26

## Contents

---

- [9-1-4](#)
- [10-2-1](#)
- [11-1-1](#)
- [14-13](#)
- [15-16](#)
- [16-9](#)

以上分別為各問題的code

以下為測試用程式主體：

```
function []= test_code_hw1()
    clear all;
    test_1();
    test_2();
    test_3();
    test_4();
    test_5();
    test_6();

end
```

## 9-1-4

---

請設計一個函數 `index2pos.m`，其用法如下：`position = index2pos(index)`；其中 `index` 是一個元素為正整數的向量，`position` 則是此函數的輸出矩陣，其列數為 `index` 元素的最大值，行數則是 `index` 的元素個數，而且當 `index(i) = j` 時，`position` 的第 `i` 行中，只有第 `j` 個元素為 1，其餘為零。

以下為 `index2pos` 函式

```
function position=index2pos(index)
% 輸入一個一維向量
% 輸出一個矩陣
% 除了每個一維向量所表示位置為1以外其他元素全部為0
    n=length(index);
    position=zeros(n);
    for i=1:n
        position(index(i),i)=1;
    end
end
```

以下為測試用函式

```
function []= test_1()
    fprintf('Here is the demo for problem: %s .\n', '9-1-4');
    a=[2 1 3 5];
    b=[2 7 1 1 5];

    disp('輸入為:');
    disp(a);
    disp('輸出為:');
    disp(index2pos(a));

    disp('輸入為:');
    disp(b);
    disp('輸出為:');
    disp(index2pos(b));

end
```

Here is the demo for problem: 9-1-4 .

輸入為：

2	1	3	5
---	---	---	---

輸出為：

0	1	0	0
1	0	0	0

0	0	1	0
0	0	0	0
0	0	0	1

輸入為：

2	7	1	1	5
---	---	---	---	---

輸出為：

0	0	1	1	0
1	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	1
0	0	0	0	0
0	1	0	0	0

## 10-2-1

Palindrome: 「迴文」 (Palindrome) 在中文當中是指倒正著念 和反著念都相同的句子，前後對稱，例如「上海自來水來自海上」。在英文當中是指正著看和反著看都相同的單字，例如「madam」。請寫一個 MATLAB 函數，來決定所給定的字串是否是「迴文」。

以下為Palindrome回文函式

```
function is_Palindrome=Palindrome(index)
% 判斷判斷函式
% 輸入一個向量
% 從頭尾依次判斷是否相等，若其中有一不相等的則視為不相等
n=length(index);
is_Palindrome='Yes,it is palindrome.';
for i=1:n
    if index(i)~=index(n+1-i)
        is_Palindrome='Nope,it is not palindrome.';
        break;
    end
% 速度優化：可將判斷n個改為判斷到一半n+1/2即可
% 但個數不多情況下影響不大
end
```

以下為測試用函式

```
function []=test_2()
fprintf('Here is the demo for problem : %s .\n','10-2-1');
a='madam';
b='homework';
c=[1 2 3 2 1];

disp('輸入為：');
disp(a);
disp('輸出為：');
disp(Palindrome(a));

disp('輸入為：');
disp(b);
disp('輸出為：');
disp(Palindrome(b));

disp('輸入為：');
disp(c);
disp('輸出為：');
disp(Palindrome(c));
%順便測試一下Cell Arrays 是否可以判斷
d=[a b a];
disp('輸入為：');
disp(d);
disp('輸出為：');
disp(Palindrome(d));
%實際測試之後發現並不是以aba做判斷，而是合併之後判斷
%若需要判斷aba可能需要用重載

end
```

Here is the demo for problem : 10-2-1 .

輸入為：

madam

```

輸出為：
Yes,it is palindrome.
輸入為：
homework
輸出為：
Nope,it is not palindrome.
輸入為：
    1      2      3      2      1

輸出為：
Yes,it is palindrome.
輸入為：
madamhomeworkmadam
輸出為：
Nope,it is not palindrome.

```

## 11-1-1

請寫一個簡短的程式getMdArray.m來建立一個3維矩陣A

以下為getMdArray建立矩陣函式

```

function [data,years,type,class] = getMdArray()
% 無輸入
% 輸出四變數[data years type class]
% data為一個4*3*2三維數字矩陣 依次存放身份/年次/班別的人數
% years為一個3*1向量 存放年次
% type為一個1*4向量 存放身份
% class為1*2向量 存放班別名稱
data=[98 94 80 5; 105 97 87 6; 121 110 89 8];
data(:,:,2)=[99 98 85 10; 113 101 87 12; 120 115 80 15];
years=[{2001},{2002},{2003}];
type=[{'大一新生'},{'學士畢業生'},{'碩士畢業生'},{'博士畢業生'}];
class=[{'資訊系'},{'電機系'}];
end

```

以下為測試用函式

```

function []=test_3()
fprintf('Here is the demo for problem : %s .\n','11-1-1');
[data,years,type,class] = getMdArray();

for i=1:2;
    disp(class(i));
    for j=1:3;
        fprintf('\t');
        disp(years(j));
        for k=1:4;
            fprintf('\t\t');
            disp(type(k));
            fprintf('\b\b');
            disp(data(j,k,i));
        end
    end
end
end

```

```

Here is the demo for problem : 11-1-1 .
'資訊系'

```

```

[2001]

'大一新生'      98

'學士畢業生'     94

'碩士畢業生'     80

'博士畢業生'      5

```

```

[2002]

'大一新生'      105

'學士畢業生'     97

```

'碩士畢業生'	87
'博士畢業生'	6
[2003]	
'大一新生'	121
'學士畢業生'	110
'碩士畢業生'	89
'博士畢業生'	8
'電機系'	
[2001]	
'大一新生'	99
'學士畢業生'	98
'碩士畢業生'	85
'博士畢業生'	10
[2002]	
'大一新生'	113
'學士畢業生'	101
'碩士畢業生'	87
'博士畢業生'	12
[2003]	
'大一新生'	120
'學士畢業生'	115
'碩士畢業生'	80
'博士畢業生'	15

## 14-13

請寫一個 MATLAB 程式找出最小的 正整數  $n$  值，使得  $A^{(n+1)}=A^n$  此時 $n$ 值為何

以下為min\_value\_times建立矩陣函式

```
function n=min_value_times(matrix)
% 輸入一矩陣matrix
% 判斷最小正整數N，使得矩陣的N次和N+1次方為相等
% 輸出N
matrix_n=matrix;
I=zeros(3)+1;
for i=1:1000;
    if (isequal(matrix^i,matrix^(i+1)))
        n=i;
        break;
    else
        matrix_n=matrix_n*matrix;
    end
end
% 速度優化：每次都計算一次n次方為0 (n^2)
% 可改善成多增開一個matrix 來回相乘避免重複計算
% 可改善成O (n)
end
```

以下為測試用函式

```
function []=test_4()
fprintf('Here is the demo for problem: %s .\n', '14-13');
M=[0.3 0.1 0.2;0.2 0.7 0.4;0.5 0.2 0.4];
n=min_value_times(M);
disp('A為: ');
```

```

format bank;
disp(M);
fprintf('最小的n值為%d.\n此時的A^n和A^{n+1}為\n',n);
format long;
disp(M^n);
format;

```

end

Here is the demo for problem : 14-13 .

A為：

0.30	0.10	0.20
0.20	0.70	0.40
0.50	0.20	0.40

最小的n值為102.

此時的A^n和A^{n+1}為

0.163934426229508	0.163934426229508	0.163934426229508
0.524590163934424	0.524590163934424	0.524590163934424
0.311475409836065	0.311475409836064	0.311475409836065

## 15-16

撰寫一函式 quadzero01.m，其輸出人的格式如下： root = quadzero01(coef) 其中 coef 是一個 3x1 的向量，代表一個一元二次方程式的係數（降冪排列），而 root 則是此方程式的根所形成的向量。若 coef(1) 不是零，則 roots 的長度為二。若 coef(1) 是零，則可能有一解（root 的長度為一）或無解（root 為空矩陣）。

以下為quadzero01建立矩陣函式

```

function root = quadzero01(coef)
% 解一元二次方程式的解
% 輸入為一1*3向量 分別表示二次、一次、和常數項的係數
% 若無解 輸出空向量
% 若非二次方程式有一次解 輸出包含一個元素的向量
% 若為二次方程式，則輸出其解（包含複數解）
switch coef(1)
case {0}
    if((coef(2)==0)&(coef(3)~=0))
        root=[];
    else
        root=[-coef(3)/coef(2)];
    end
otherwise
    root=[coef(2) coef(2)].^2-(4*[coef(1) coef(1)].*[coef(3) coef(3)]);%b^2-4ac
    root=root.^(1/2);%sqrt(b^2-4ac)
    root=root.*[1 -1];%+-sqrt(b^2-4ac)
    root=root-coef(2);%-b+-sqrt(b^2-4ac)
    root=root/(2*coef(1));%(-b+-sqrt(b^2-4ac))/2a
end

```

end

以下為測試用函式

```

function []=test_5()
fprintf('Here is the demo for problem : %s .\n','15-16');
a=[0 0 1];
b=[0 2 4];
c=[1 -2 1];
d=[1 -5 6];

disp('係數：');
disp(a);
disp('根為：');
disp(quadzero01(a));

disp('係數：');
disp(b);
disp('根為：');
disp(quadzero01(b));

disp('係數：');
disp(c);
disp('根為：');
disp(quadzero01(c));

```

```

disp('係數：');
disp(d);
disp('根為：');
disp(quadzero01(d));

```

end

Here is the demo for problem : 15-16 .

係數：

0      0      1

根為：

係數：

0      2      4

根為：

-2

係數：

1      -2      1

根為：

1      1

係數：

1      -5      6

根為：

3      2

## 16-9

給定一向量 A，請寫一段程式 usef01.m，利用 if-then-else 指令來依元素值不同 而印出不同訊息

以下為usef01建立矩陣函式

```

function []=usef01(index)
%    輸入一個向量，依次呼叫判斷函數judge
%    依次印出結果
for k=1:length(index)
    fprintf('A(%d) = ',k);
    judge(index(k));
end

end

function []=judge(number)
%    判斷輸入number為複數或是正負零，並印出其結果
if ~isreal(number)
    fprintf('%f+%fi是複數\n',real(number),imag(number));
else if (number==0)
    fprintf('%f是零\n',number);
else
    switch (number/abs(number))
        case {1}
            fprintf('%f是正數\n',number);
        case {-1}
            fprintf('%f是負數\n',number);
    end
end
end

end

```

以下為測試用函式

```

function []=test_6()
fprintf('Here is the demo for problem : %s .\n','16-9');
A=randn(20,1)+(rand(20,1)>0.7)*sqrt(-1);

disp('輸入為：');
disp(A);

```

```
disp('輸出為：');  
useIf01(A);
```

```
end
```

Here is the demo for problem : 16-9 .

輸入為：

```
1.4193 + 1.0000i  
0.2916 + 0.0000i  
0.1978 + 0.0000i  
1.5877 + 0.0000i  
-0.8045 + 0.0000i  
0.6966 + 0.0000i  
0.8351 + 0.0000i  
-0.2437 + 0.0000i  
0.2157 + 1.0000i  
-1.1658 + 0.0000i  
-1.1480 + 0.0000i  
0.1049 + 0.0000i  
0.7223 + 1.0000i  
2.5855 + 0.0000i  
-0.6669 + 1.0000i  
0.1873 + 0.0000i  
-0.0825 + 1.0000i  
-1.9330 + 0.0000i  
-0.4390 + 0.0000i  
-1.7947 + 0.0000i
```

輸出為：

```
A(1) = 1.419310+(1.000000)i是複數  
A(2) = 0.291584是正數  
A(3) = 0.197811是正數  
A(4) = 1.587699是正數  
A(5) = -0.804466是負數  
A(6) = 0.696624是正數  
A(7) = 0.835088是正數  
A(8) = -0.243715是負數  
A(9) = 0.215670+(1.000000)i是複數  
A(10) = -1.165844是負數  
A(11) = -1.147953是負數  
A(12) = 0.104875是正數  
A(13) = 0.722254+(1.000000)i是複數  
A(14) = 2.585491是正數  
A(15) = -0.666891+(1.000000)i是複數  
A(16) = 0.187331是正數  
A(17) = -0.082494+(1.000000)i是複數  
A(18) = -1.933023是負數  
A(19) = -0.438966是負數  
A(20) = -1.794679是負數
```