

1 Ledning till uppgift 1

I MATLAB finns två sätt att avgöra om en matris innehåller ett visst element. Antingen använder man `for`-loopar och löper igenom elementen i matrisen, eller så använder man vektorisering.

Med `for`-loop

Låt `A` vara matrisen i exemplet

```
>> A
A =  0  1  1  1  1
     0  1  1  1  1
     1  1  1  1  1
     1  1  1  2  2
     1  1  2  0  0
```

Testa anropen `A(1,1) == 0` och `A(1,1) == 2` (skriv in dem och testa i MATLAB). Det första anropet är sant (elementet på första raden i första kolumnen är 0), och MATLAB svarar 1 för `true`. Det andra anropet är falskt (elementet på första raden i första kolumnen är inte 2) och MATLAB svarar 0 för `false`.

Elementen i en matris har två index, ett för raden och ett för kolumnen. När man kontrollerar alla element i en matris använder man dubbla `for`-loopar för att stega igenom alla index. Låt `A` vara matrisen i exemplet. Testa att skriva in följande kod i MATLAB:

```
for i = 1:5
    for j = 1:5
        disp(A(i,j))
    end
end
```

I vilken ordning skrivs elementen i `A` ut (radvis eller kolumnvis)?

Vi byter ut `disp(A(i,j))` till `if`-satsen

```
if A(i,j) == 2
    return;
end
```

och får

```
for i = 1:5
    for j = 1:5
        if A(i,j) == 2
            return;
        end
    end
end
```

(**return**-satsen avbryter loopen). Låt **A** vara matrisen i exemplet och testkör loopen ovan. Vilka värden har **i** och **j** när loopen avbryts?

Skriv nu en funktion (en extern **function**) som returnerar **true** om en $n \times n$ matris innehåller minst en 2:a. Om matrisen saknar 2:or ska funktionen returnera **false**. Låt funktionen ha matrisen som inparameter.

```
function t = finns2(A)
    [m,n] = size(A);
    t = false;
    for i = 1:n
        for j = 1:n
            ...
```

Med vektorisering

Om man inte vill använda **for**-loopar för att avgöra om ett visst element finns i en matris, kan man istället använda vektorisering. Låt **A** vara matrisen i exemplet

```
>> A
A =  0  1  1  1  1
     0  1  1  1  1
     1  1  1  1  1
     1  1  1  2  2
     1  1  2  0  0
```

Testa anropet

```
>> A == 2
```

MATLAB svarar med en matris som har 1:or (**true**) på de platser det finns 2:or i **A**, 0 (**false**) på de platser som inte är 2.

Det finns ett kommando som heter **any** som används för att avgöra om det finns några värden i en matris som har värdet sant (**true**). Kommandot opererar på kolumnerna i matrisen. Låt **A** vara matrisen i exemplet. Anropet

```
>> any(A == 2)
ans =  0  0  1  1  1
```

svarar **false** för de första två kolumnerna och **true** för de sista 3. Det finns inga 2:or i de första två kolumnerna, men de sista tre innehåller varsin 2:a. Om man vill att **any** ska svara med bara ett värde kan man antingen anropa kommandot en gång till

```
>> any(any(A == 2))
```

eller göra om elementen i **A** till en vektor och göra anropet

```
>> any(A(:) == 2)
```