

## En programmeringsövning

Vi ska skriva ett program som simulerar en skogsbrand, dvs. hur en eld sprider sig över ett område under en viss tid. Att simulera hur en skogsbrand sprider sig är naturligtvis väldigt komplicerat, därför kommer vår modell att vara förenklad – så den här laborationen avser framförallt att träna programmering.

Ett sätt att simulera ett brandförlopp över ett område över en viss tid är att dela in området i mindre delar (celler). Varje cell motsvarar en viss area i skogen, och varje cell har en viss status (tex. brinner, brinner inte, kan inte brinna). Med hjälp av regler beskrivs hur en cell byter från en status till en annan över tid.

**Vår modell** Vi tänker oss att skogsområdet är uppdelat i  $n \times n$  celler. Varje cell motsvarar en liten area i skogen, och varje cell kan anta en av följande värden (status)

- 0 - indikerar att det finns ingenting i cellen som kan börja brinna (cellen består tex. av sand, eller träd som redan har brunnit upp).
- 1 - indikerar att cellen brinner inte just nu (men det finns tex. träd i cellen som skulle kunna börja brinna).
- 2 - indikerar att cellen brinner.

Exempel: Om vi låter  $n = 5$  och delar in skogsområdet i  $5 \times 5$  celler enligt:

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

Tre av cellerna brinner (motsvaras av de rutor som innehåller siffran 2), fyra av cellerna kan inte börja brinna (de rutor som innehåller 0) och resten av cellerna brinner inte.

I MATLAB kan vi enkelt representera området med en  $5 \times 5$  matris, där varje element innehåller respektive cells status (dvs. 0, 1 eller 2).

```
>> n = 5;
>> A = ones(n);
>> A(1:2,1) = 0; A(n,n-1:n) = 0;
>> A(n,3) = 2; A(n-1,n-1:n) = 2
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
```

**Uppgift 1.** Skriv en funktion i MATLAB som returnerar **true** om en  $n \times n$ -matris innehåller minst en 2:a. Om matrisen saknar tvåor ska funktionen returnera **false**. Låt funktionen ha en parameter, matrisen A.

**Regler för hur branden sprider sig över området** Om en cell i området brinner, kommer elden att sprida sig till omgivande celler med en viss sannolikhet efter en viss tid (Hur elden sprider sig beror ju på vindhastighet, hur torrt det är, om det finns något i omkringliggande celler som kan börja brinna, etc).

I vår (mycket enkla) modell gäller följande regler:

- Om en cell har status 0 (dvs det finns inget i cellen som kan börja brinna) - kommer den alltid att ha status 0.
- Om en cell har status 2 (dvs brinner) kommer den i nästa tidssteg att ha brunnit upp och få status 0.
- Om en cell har status 1 och minst en granncell brinner (dvs. har status 2) kommer cellen att börja brinna med sannolikhet  $p$ , där  $0 \leq p \leq 1$ . Med grannceller menas celler som har en kant eller hörn gemensamt med cellen.

Vi tittar på vårt exempel igen. De celler som har status 0 kommer att ha status 0 även i nästa tidssteg.

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

De celler som har status 2 kommer att i nästa tidssteg att bli 0.

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

De celler som har status 1 och har minst en brinnande granne kommer eventuellt (med sannolikhet  $p$ ) bli 2 i nästa tidssteg.

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

En funktion som beräknar antalet brinnande grannar till en cell i en  $n \times n$  matris (antalet grannar som har värdet 2) kan se ut så här:

```
function antal = antalgrannar(A,i,j,n)
    igrannar = max(1,i-1):min(n,i+1);
    jgrannar = max(1,j-1):min(n,j+1);
    if A(i,j) == 2
        antal = sum(sum(A(igrannar, jgrannar)==2))-1;
    else
        antal = sum(sum(A(igrannar, jgrannar)==2));
    end
```

Funktionen har 4 parametrar, matrisen  $A$ , rad- och kolumnindex för cellen  $(i, j)$  och värdet på  $n$ . Om vi anropar funktionen med  $5 \times 5$  matrisen  $A$  i exemplet ovan kan det se ut så här:

```
>> antalgrannar(A,1,1,5)
ans = 0
>> antalgrannar(A,4,3,5)
ans = 2
```

## Uppgift 2.

- Skriv en sekvens i MATLAB som uppdaterar matrisen  $A$  i exemplet ovan ett tidssteg. Använd funktionen `antalgrannar` om du behöver. Låt sannolikheten  $p$  för att en cell ska börja brinna vara 0.5. (Använd `rand` för att få till sannolikheten<sup>1</sup>).
- Skriv sekvensen så att matrisen uppdateras flera tidssteg, ända tills det inte finns celler kvar som kan börja brinna.

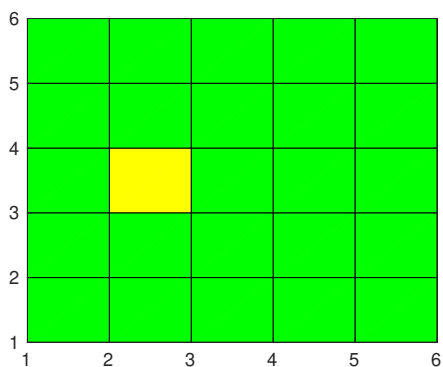
Matlabkoden nedan ritar  $5 \times 5$  gröna rutor i ett figurfönster.

```
figure(1); hold on;
n = 5;
for i = 1:n
    for j = 1:n
        F(i,j)=fill([i i+1 i+1 i],[j j j+1 j+1],'green');
    end
end
```

När man använder `fill` på detta sätt kommer  $F$  att referera till de olika rutorna i figuren. Anropet  $F(2,3)$  t.ex. refererar till rutan med hörnkoordinaterna  $(2,3)$ ,  $(3,3)$ ,  $(3,4)$ ,  $(2,4)$  och anropet

```
F(2,3).FaceColor = 'yellow';
```

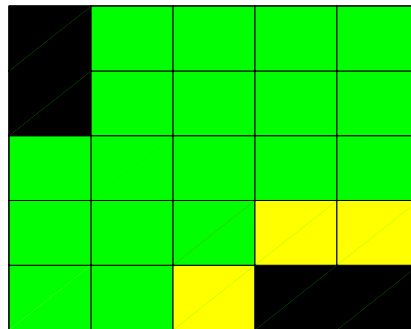
färgar rutan gul.



<sup>1</sup>Anropet `rand < 0.5` kommer att vara sant i 50% av fallen.

Vi tittar på exemplet ovan igen. Området kan presenteras grafiskt.

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



För att rita figuren till höger har man använt `fill`. Cellerna som inte kan brinna har ritats svarta, de som brinner gula och de som inte brinner är gröna.

**Uppgift 3.** Rita figuren till höger ovan i MATLAB. (Om du vill ta bort koordinataxlarna i figuren kan du använda `axis off`).

**Uppgift 4.** Gör om uppgift 2. Illustrera förloppet grafiskt, dvs rita alla de olika matriserna i en figur. Använd `fill` enligt ovan.

Följande sekvens i MATLAB ritar  $5 \times 5$  gröna rutor i ett figurfönster och låter användaren klicka ut svarta rutor i figuren, en ruta i taget.

```
figure(1); hold on;
n = 5;
for i = 1:n
    for j = 1:n
        F(i,j)=fill([i i+1 i+1 i],[j j j+1 j+1],'green');
    end
end

while 1
    [x,y,knapp] = ginput(1);
    if knapp ~= 1
        break;
    end
    x = floor(x); y = floor(y);
    F(x,y).FaceColor = 'black';
end
```

**Uppgift 5.** Skriv ett program som skapar en skog med  $25 \times 25$  celler, gröna och ev. några svarta och låt användaren markera vilka celler som ska brinna (använd `ginput`). Illustrera sedan förloppet tills hela skogen brunnit upp, eller tills det inte längre brinner någonstans.

Använd `fill` enligt ovan för att illustrera området grafiskt. Använd ev. `pause` för att lägga in en kort paus mellan uppritning av de olika tidstegen. Låt  $p$  (sannolikheten för att en cell ska börja brinna) vara 0.9. (Om man inte använder `pause` behöver man ibland tvinga MATLAB att rita med kommandot `drawnow`).