



Computational Thinking

en

Python

Python

Programmeertaal

Algoritme → Computerprogramma

Programmeeromgeving → IDLE
→ TI-Nspire software

Ontworpen en ontwikkeld door Guido van Rossum (NL) en genoemd naar Monty Python



Python

Turtle graphics

Algorithmen

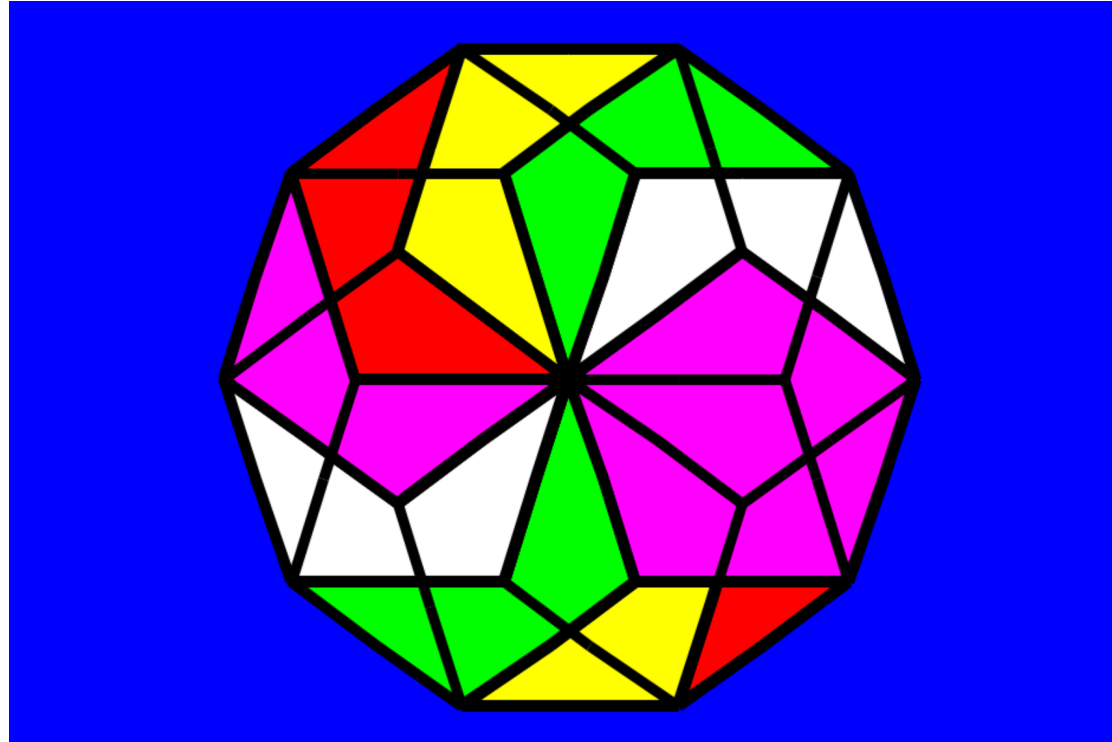
Python

Turtle graphics

Algorithmen

Python

Turtle Graphics



Turtle Module

Basis commando's:

`t.forward()`

`t.backward()`

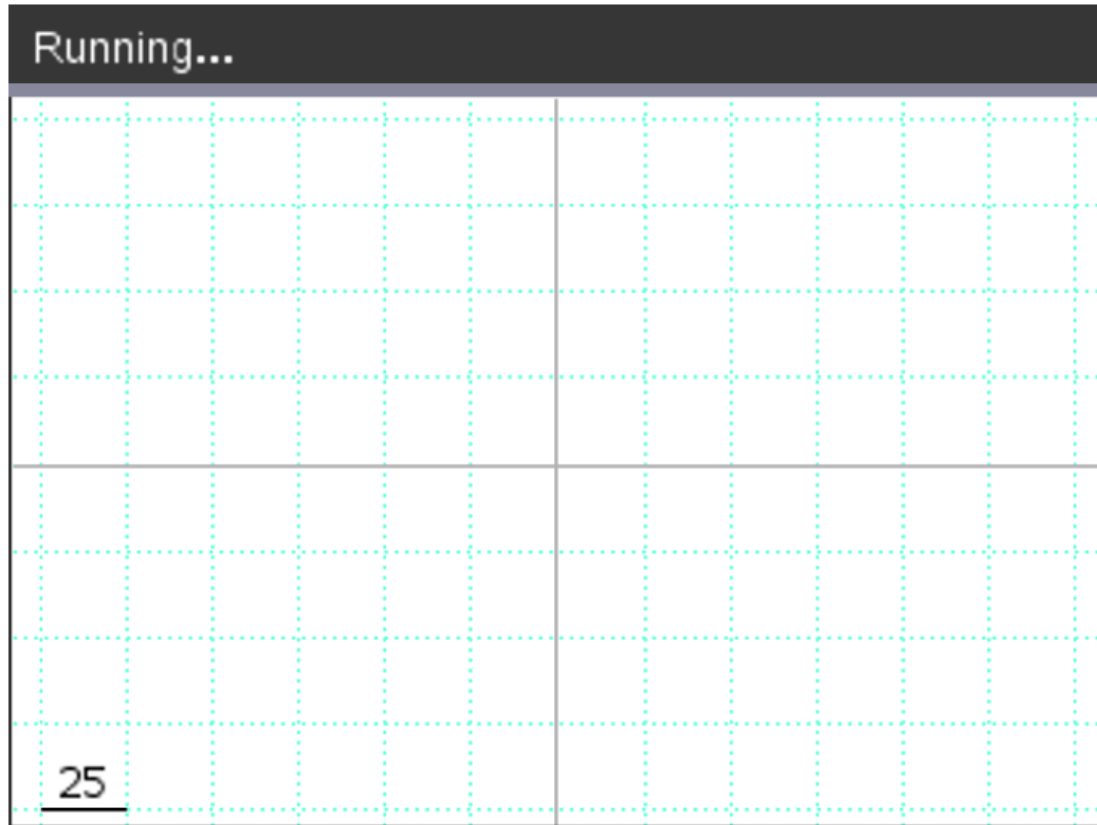
`t.left()`

`t.right()`

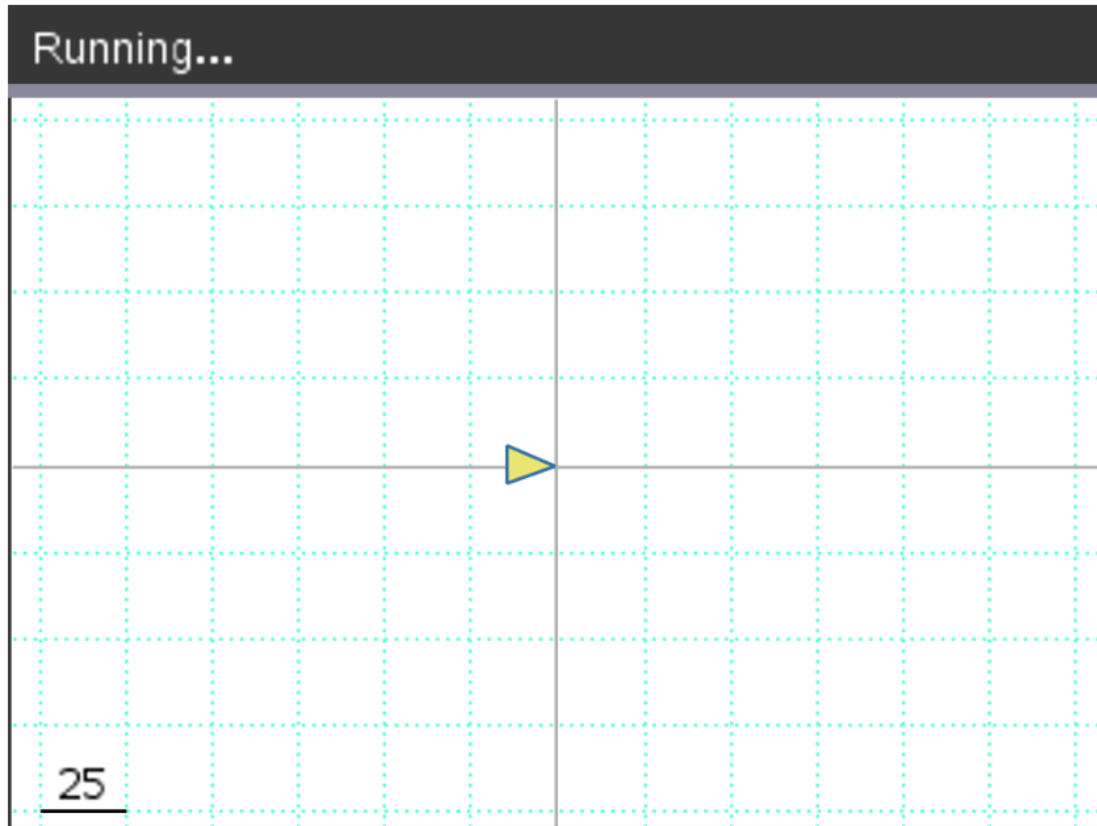
`t.penup()`

`t.pendown()`

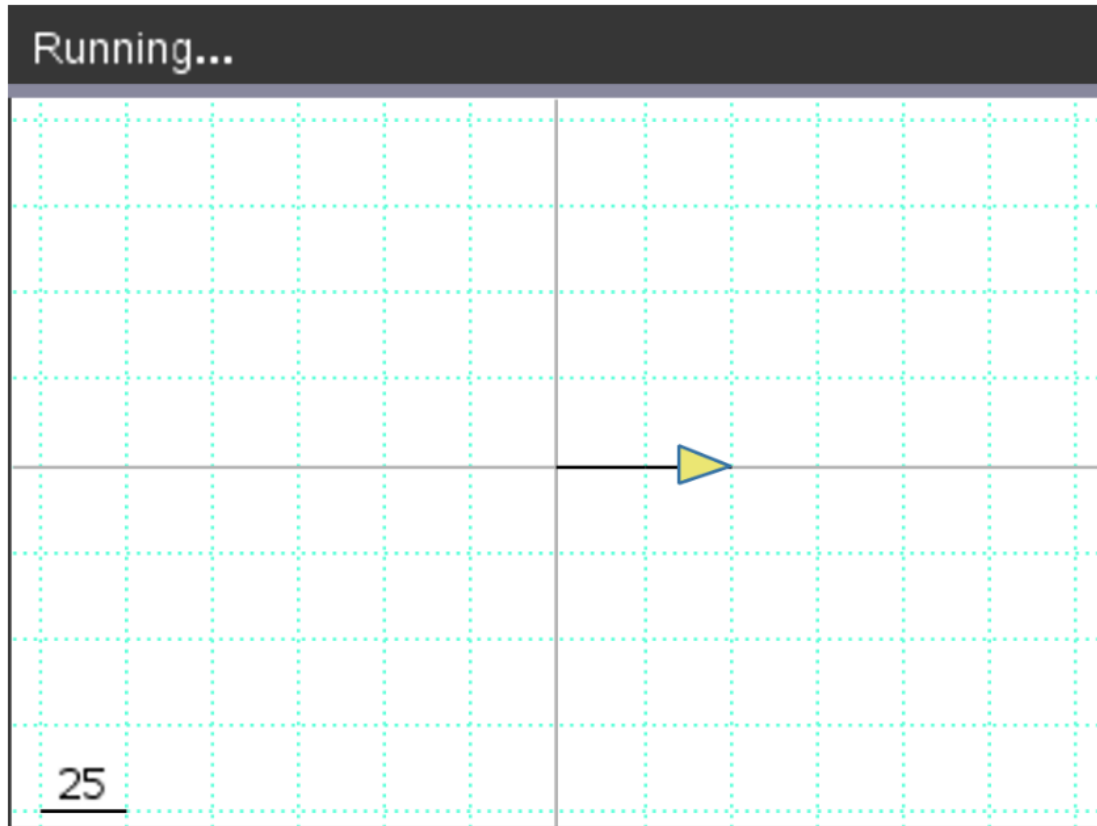
Turtle Module



Turtle Module

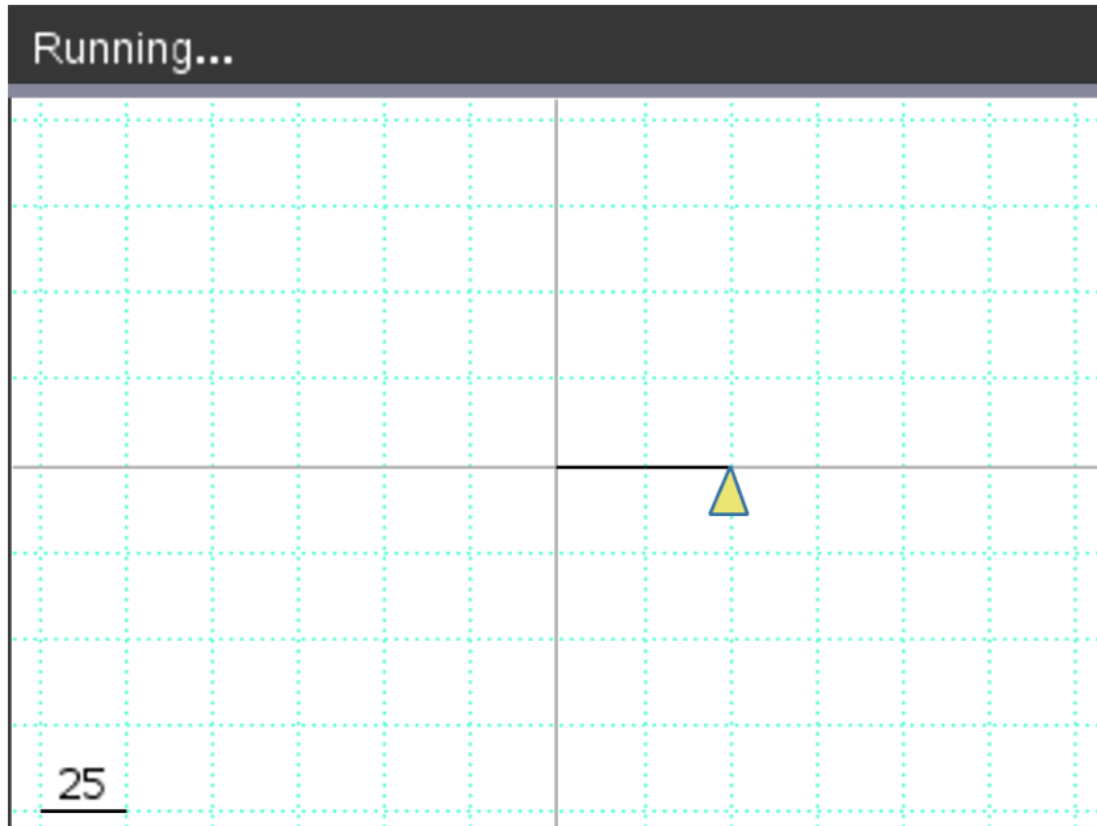


Turtle Module



`t.forward(50)`

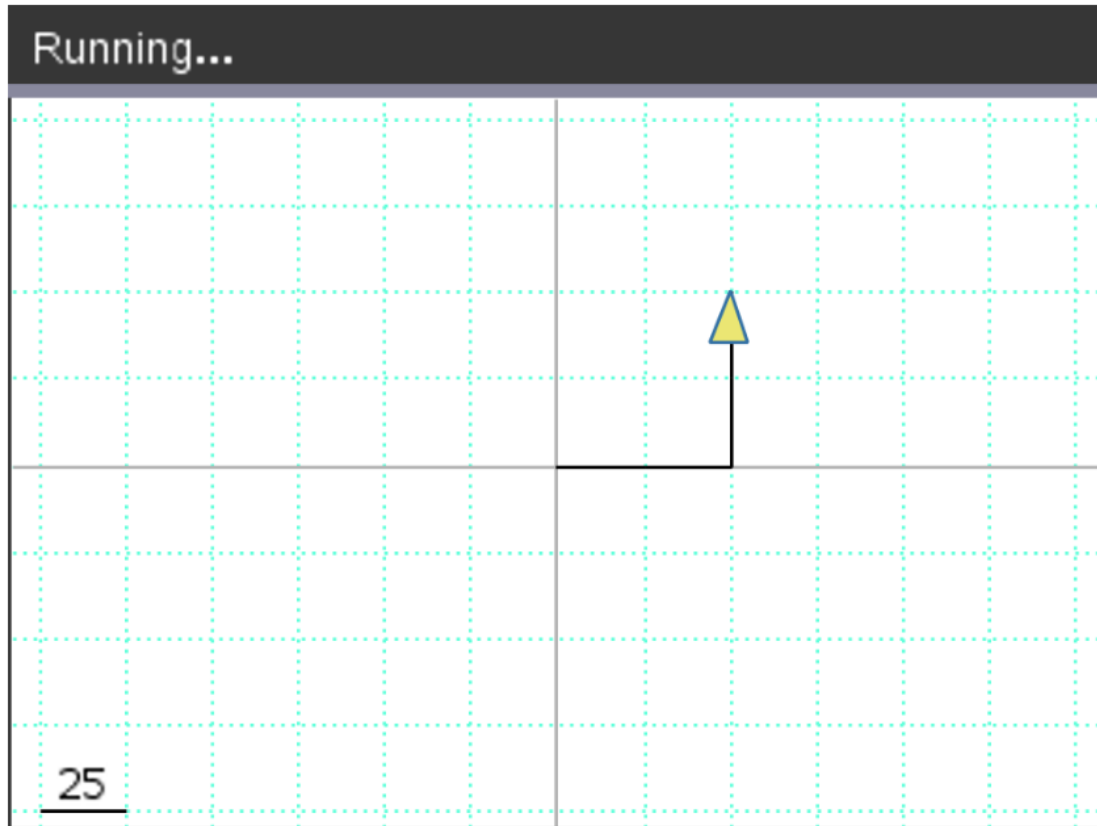
Turtle Module



```
t.forward(50)
```

```
t.left(90)
```

Turtle Module

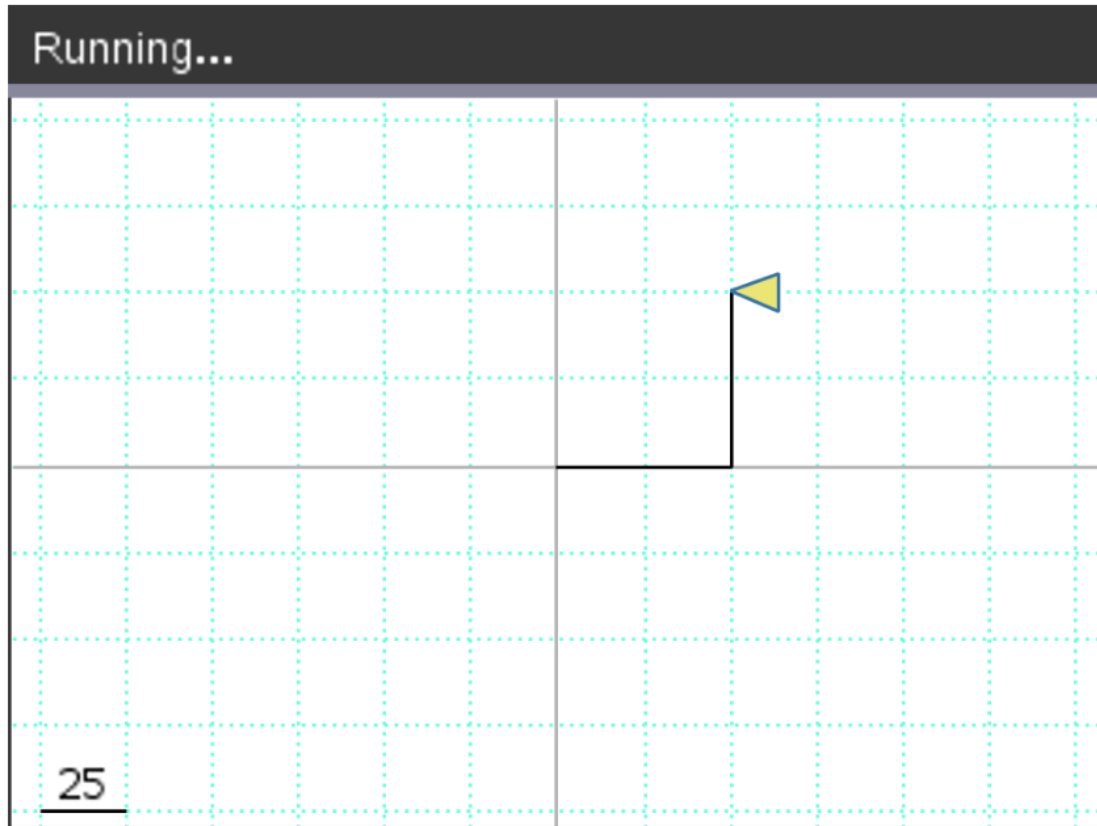


```
t.forward(50)
```

```
t.left(90)
```

```
t.forward(50)
```

Turtle Module



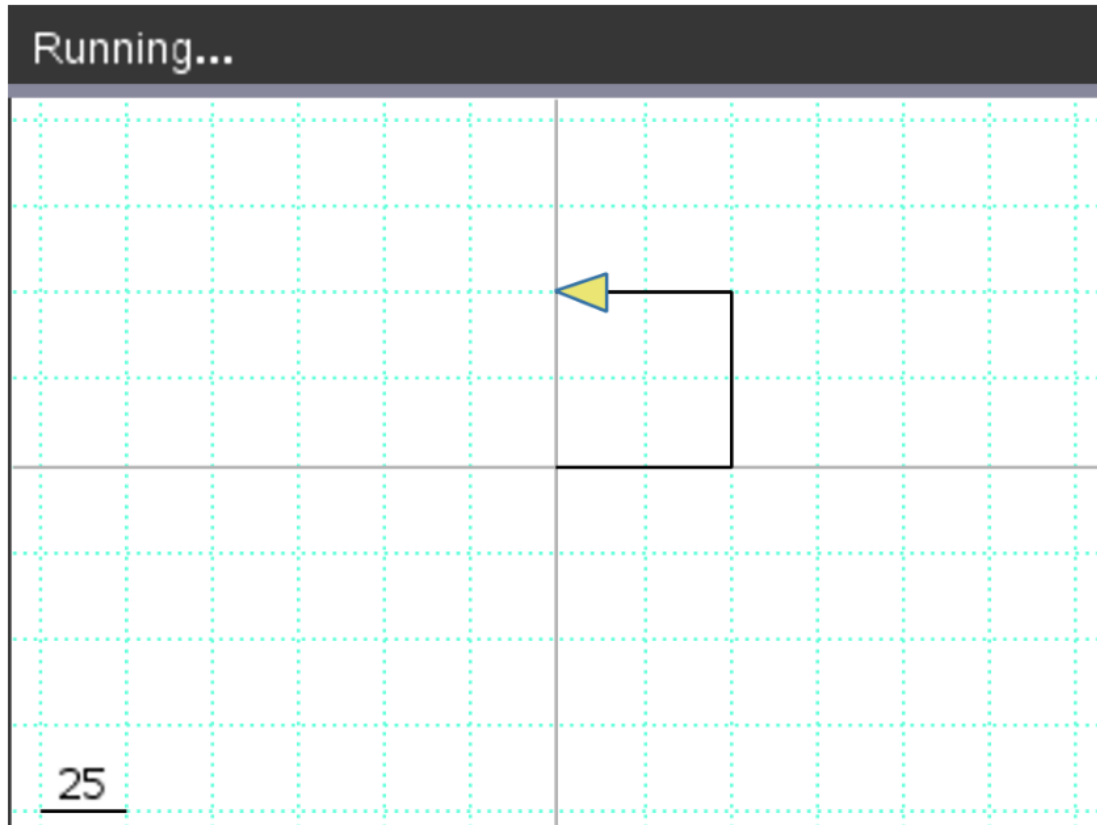
```
t.forward(50)
```

```
t.left(90)
```

```
t.forward(50)
```

```
t.left(90)
```

Turtle Module



```
t.forward(50)
```

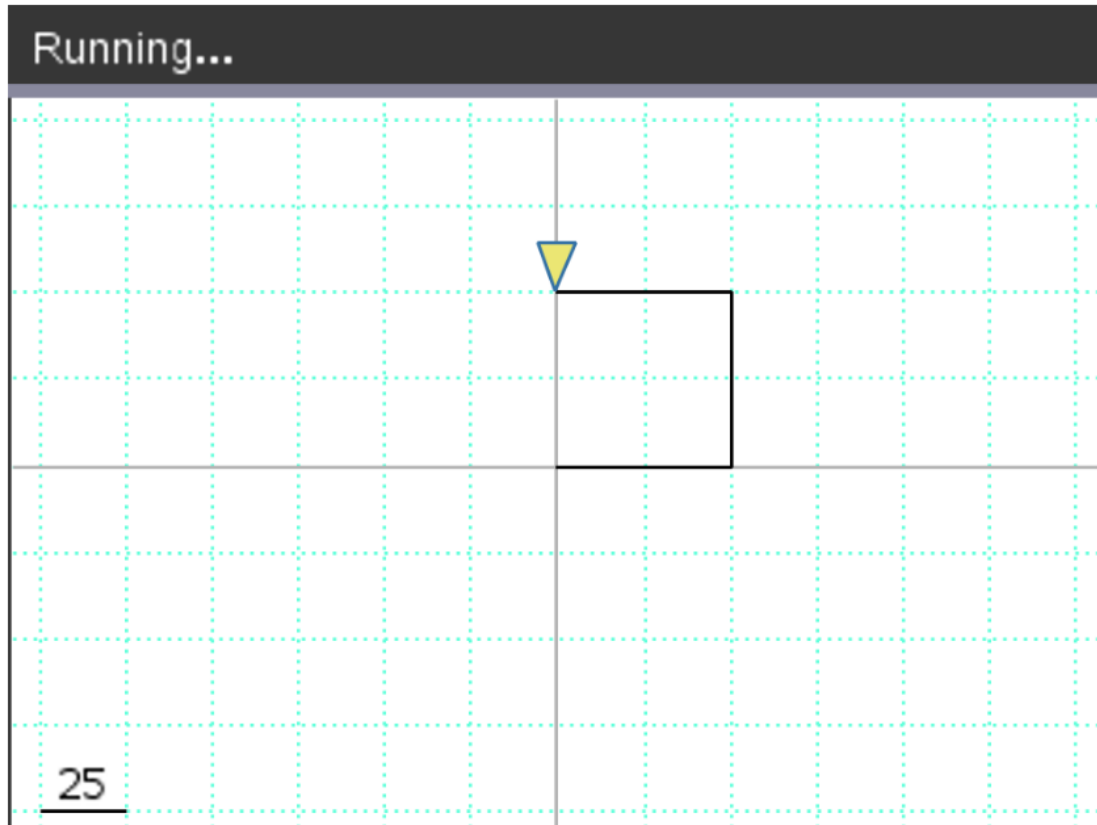
```
t.left(90)
```

```
t.forward(50)
```

```
t.left(90)
```

```
t.forward(50)
```

Turtle Module



```
t.forward(50)
```

```
t.left(90)
```

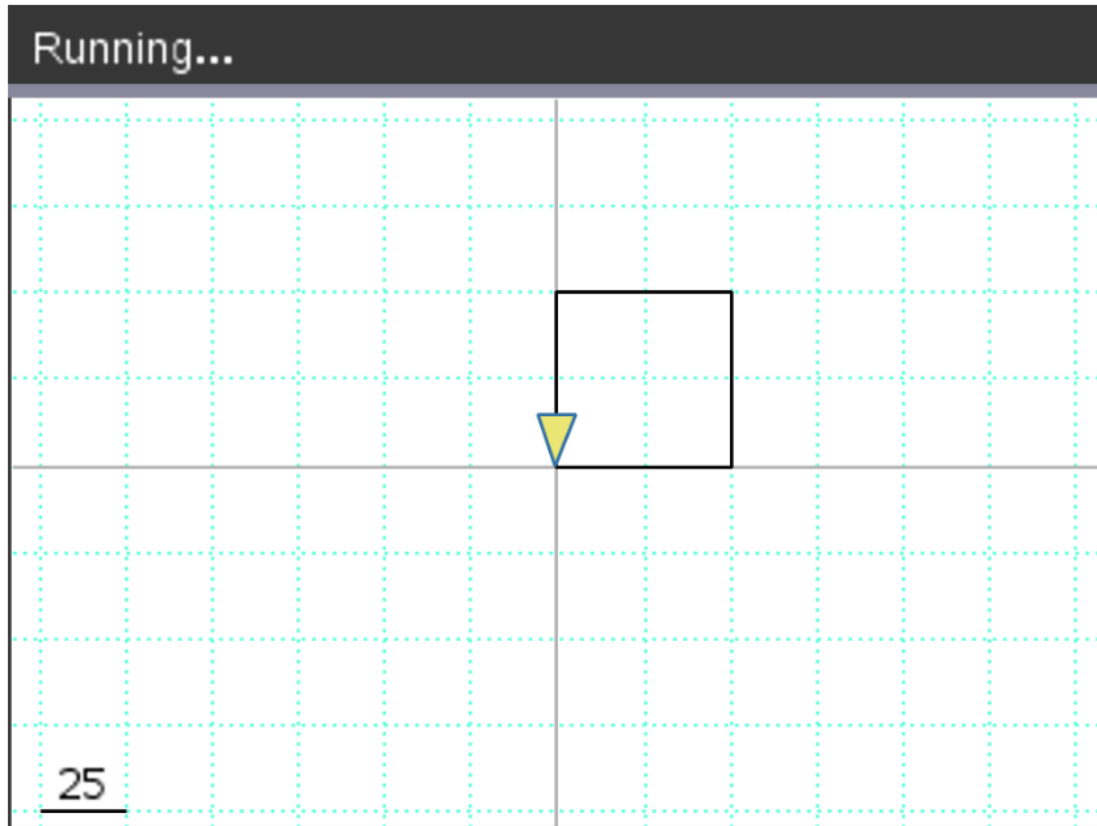
```
t.forward(50)
```

```
t.left(90)
```

```
t.forward(50)
```

```
t.left(90)
```

Turtle Module



```
t.forward(50)
```

```
t.left(90)
```

```
t.forward(50)
```

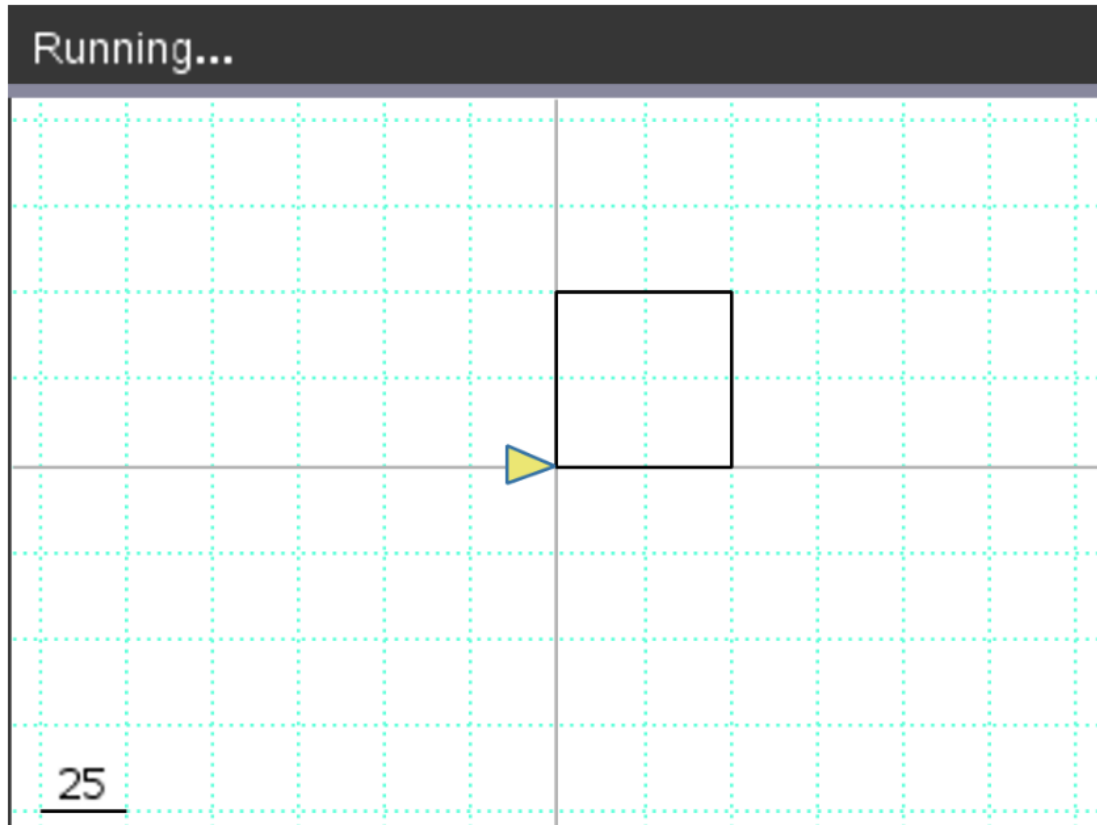
```
t.left(90)
```

```
t.forward(50)
```

```
t.left(90)
```

```
t.forward(50)
```


Turtle Module



```
t.forward(50)  
t.left(90)
```

```
t.forward(50)  
t.left(90)
```

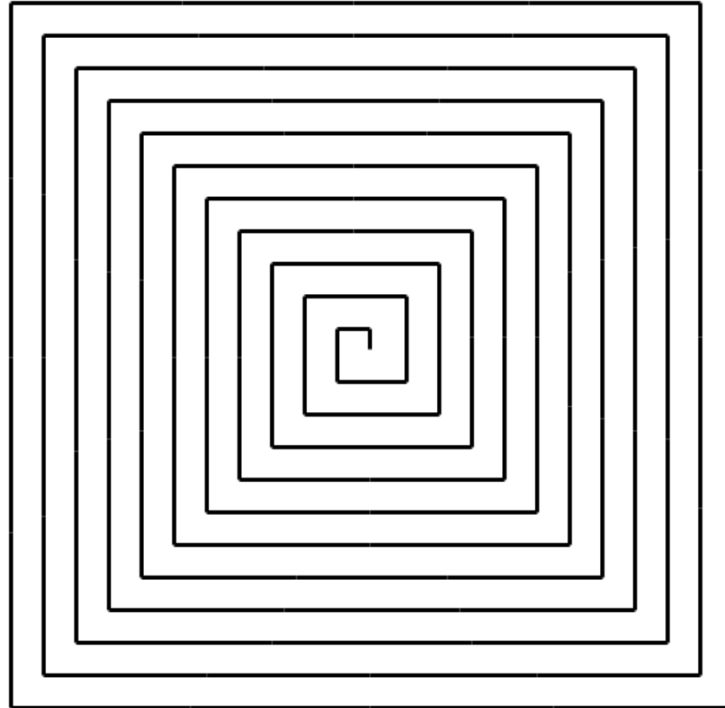
```
t.forward(50)  
t.left(90)
```

```
t.forward(50)  
t.left(90)
```

```
for i in range(4):  
    ♦♦ t.forward(50)  
    ♦♦ t.left(90)
```

Turtle Module

Spiralen



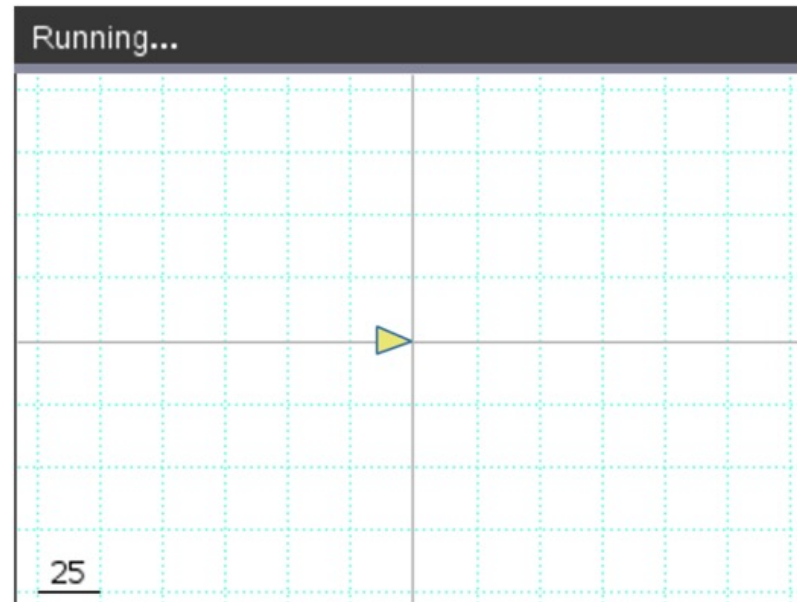
```
from turtle import *  
t=Turtle()  
t.hideturtle()  
t.speed(10)
```

```
for i in range(45):  
    t.forward(4*i)  
    t.left(90)
```

Turtle Module

Regelmatige veelhoeken

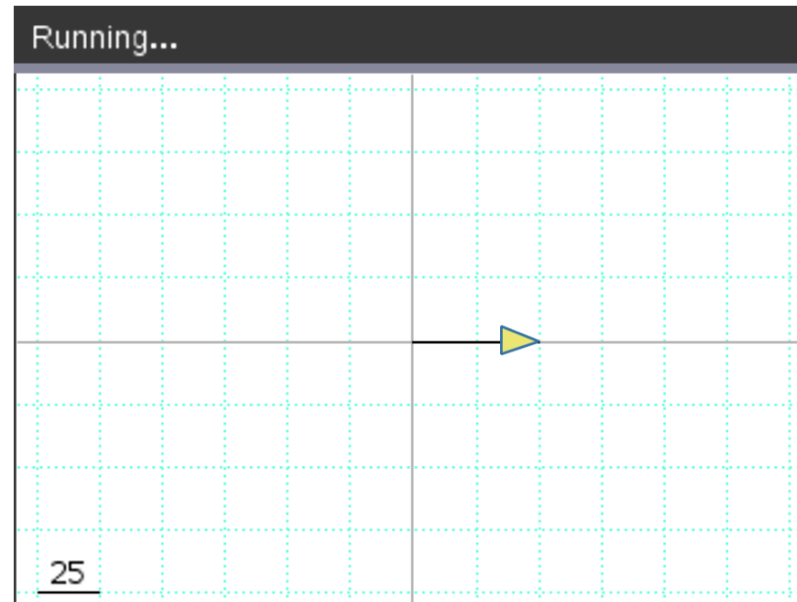
```
def driehoek():  
    for i in range(3):  
        t.forward(50)  
        t.left(120)
```



Turtle Module

Regelmatige veelhoeken

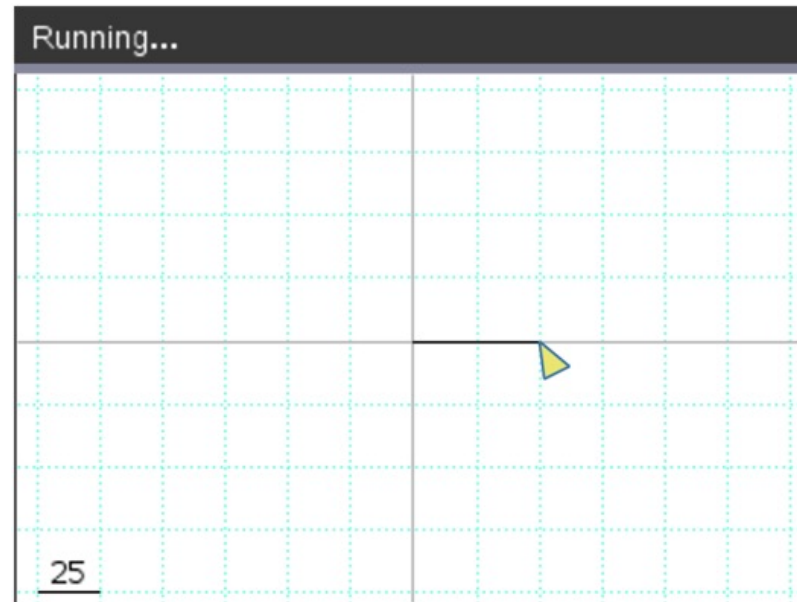
```
def driehoek():  
    for i in range(3):  
        t.forward(50)  
        t.left(120)
```



Turtle Module

Regelmatige veelhoeken

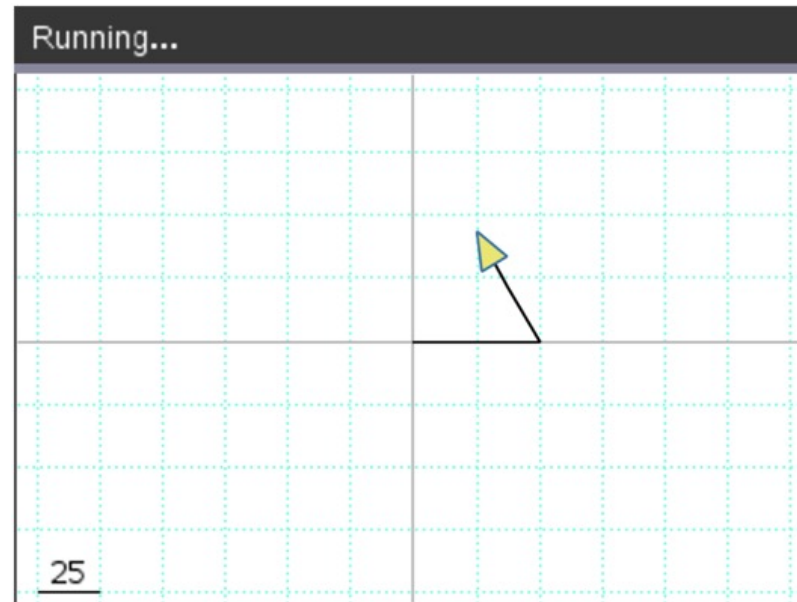
```
def driehoek():  
    for i in range(3):  
        t.forward(50)  
        t.left(120)
```



Turtle Module

Regelmatige veelhoeken

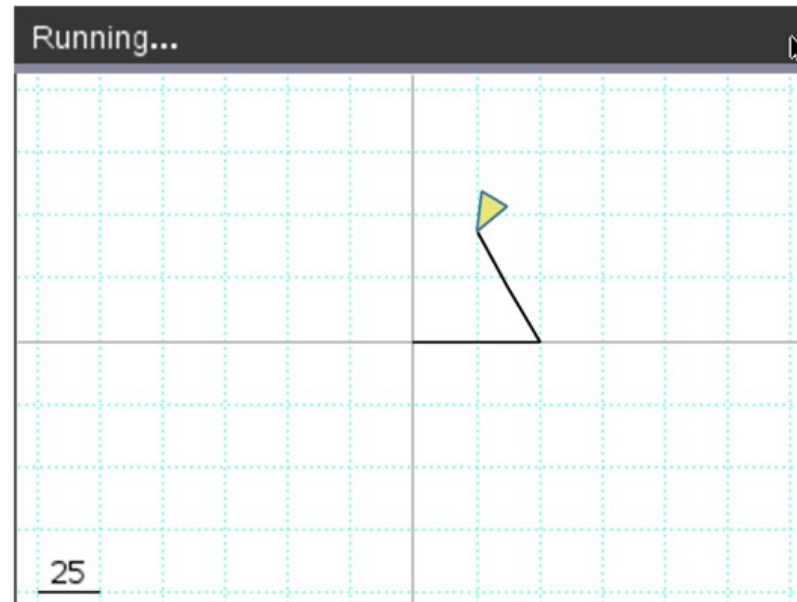
```
def driehoek():  
    for i in range(3):  
        t.forward(50)  
        t.left(120)
```



Turtle Module

Regelmatige veelhoeken

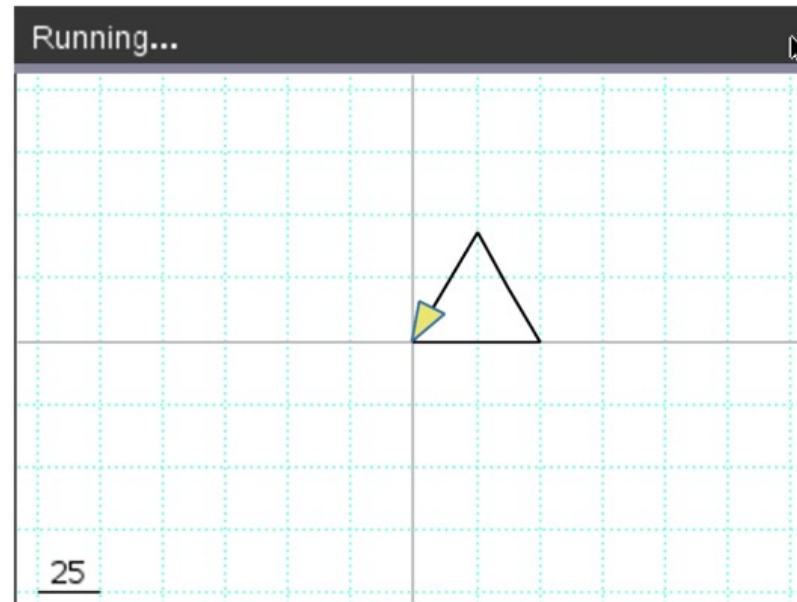
```
def driehoek():  
    for i in range(3):  
        t.forward(50)  
        t.left(120)
```



Turtle Module

Regelmatige veelhoeken

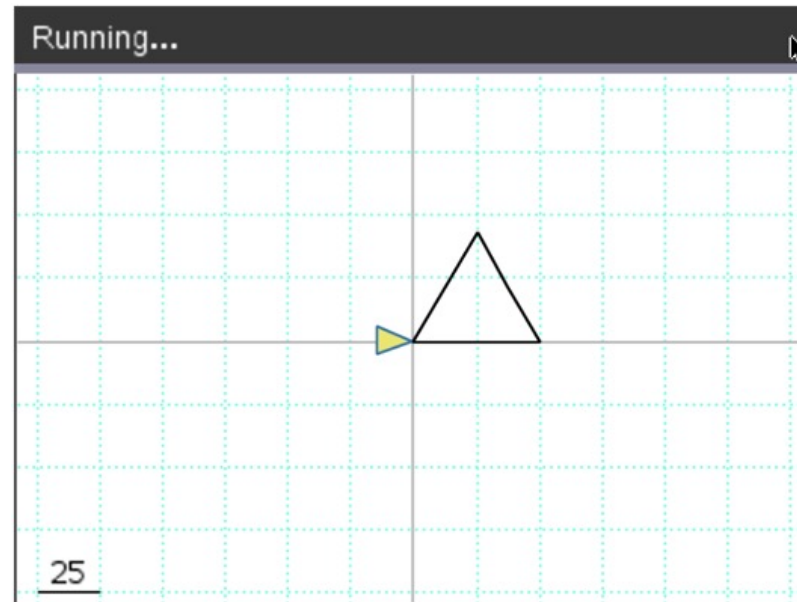
```
def driehoek():  
    for i in range(3):  
        t.forward(50)  
        t.left(120)
```



Turtle Module

Regelmatige veelhoeken

```
def driehoek():  
    for i in range(3):  
        t.forward(50)  
        t.left(120)
```



Turtle Module

Regelmatige veelhoeken

```
def driehoek():  
    for i in range(3):  
        t.forward(50)  
        t.left(120)
```

```
def vierhoek():  
    for i in range(4):  
        t.forward(50)  
        t.left(90)
```

```
def vijfhoek():  
    for i in range(5):  
        t.forward(50)  
        t.left(72)
```

```
def zeshoek():  
    for i in range(6):  
        t.forward(50)  
        t.left(60)
```

Turtle Module

Regelmatige veelhoeken

```
def driehoek():  
    for i in range(3):  
        t.forward(50)  
        t.left(120)
```

```
def vierhoek():  
    for i in range(4):  
        t.forward(50)  
        t.left(90)
```

```
def vijfhoek():  
    for i in range(5):  
        t.forward(50)  
        t.left(72)
```

```
def zeshoek():  
    for i in range(6):  
        t.forward(50)  
        t.left(60)
```

Turtle Module

Regelmatige veelhoeken

```
def driehoek():  
    for i in range(3):  
        t.forward(50)  
        t.left(120)
```

```
def vierhoek():  
    for i in range(4):  
        t.forward(50)  
        t.left(90)
```

```
def vijfhoek():  
    for i in range(5):  
        t.forward(50)  
        t.left(72)
```

```
def zeshoek():  
    for i in range(6):  
        t.forward(50)  
        t.left(60)
```

```
def veelhoek(n):  
    for i in range(n):  
        t.forward(50)  
        t.left(360/n)
```

Turtle Module

Regelmatige veelhoeken

```
def veelhoek(n):  
    for i in range(n):  
        t.forward(50)  
        t.left(360/n)
```

Turtle Module

Regelmatige veelhoeken

```
def veelhoek(n, a):  
    for i in range(n):  
        t.forward(a)  
        t.left(360/n)
```

Turtle Module

Settings

`t.hideturtle()`

`t.hidegrid()`

`t.speed()`

`t.pencolor()`

`t.fillcolor()`

Python

Turtle

Spiralen

Spirolateralen

Algorithmen

Turtle Module

Turtle Module

Spirolateralen

Stappen: [1, 2, 3, 4, 5]

Hoek: 90° (naar links)

Turtle Module

Spirolateralen

Terug

Algoritmes

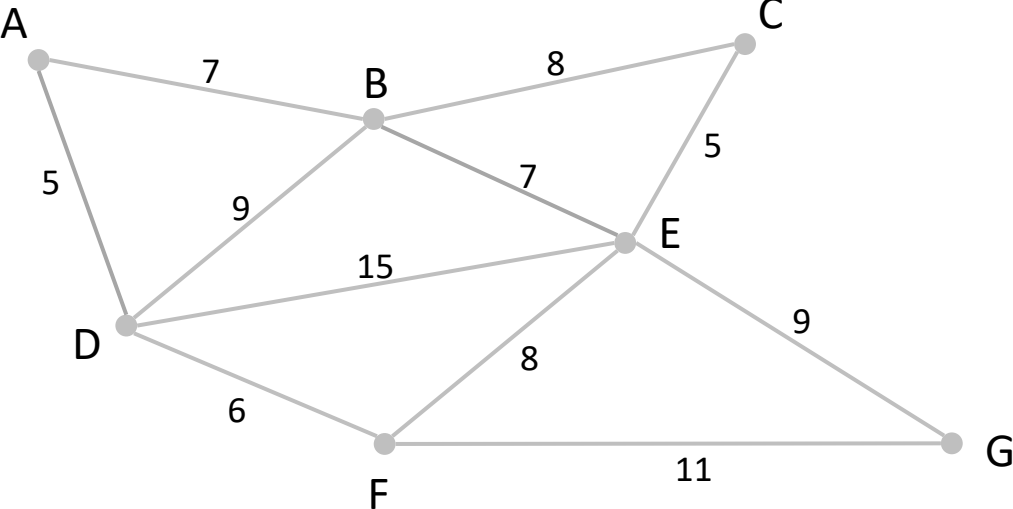
Prim algoritme

K-means algoritme

Hanoi

Langton

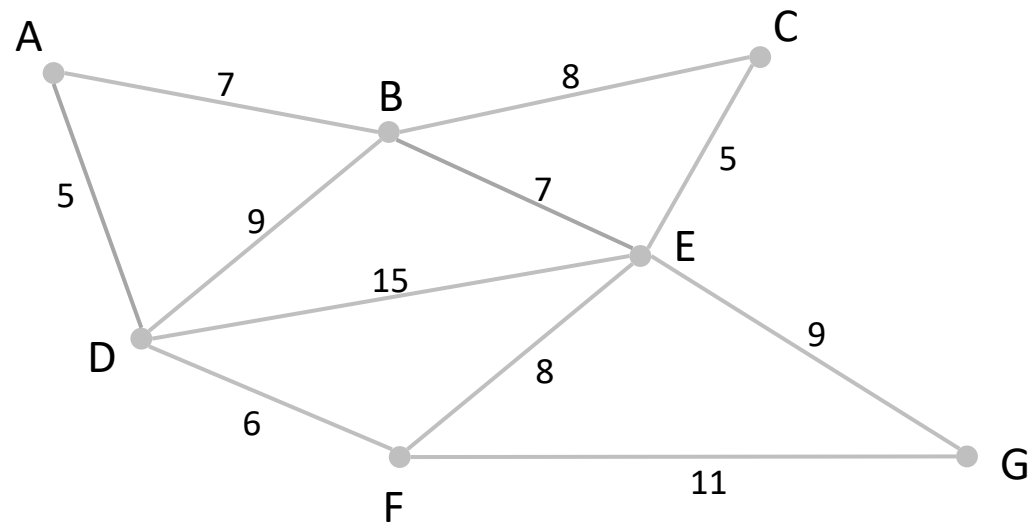
Prim algoritme



Prim algoritme

1. Start met een willekeurig punt P, stop dit punt in een lijst met verbonden punten en stop alle andere punten in een lijst met niet-verbonden punten.
2. Zoek voor elk punt P uit de lijst met verbonden punten het punt Q uit de lijst met niet-verbonden punten waarvoor de afstand tot P minimaal is.
Verbind P met Q.
Verwijder Q uit de lijst met niet-verbonden punten.
Voeg Q toe aan de lijst met verbonden punten.
3. Herhaal stap 2 voor alle verbonden punten tot de lijst met niet-verbonden punten leeg is.

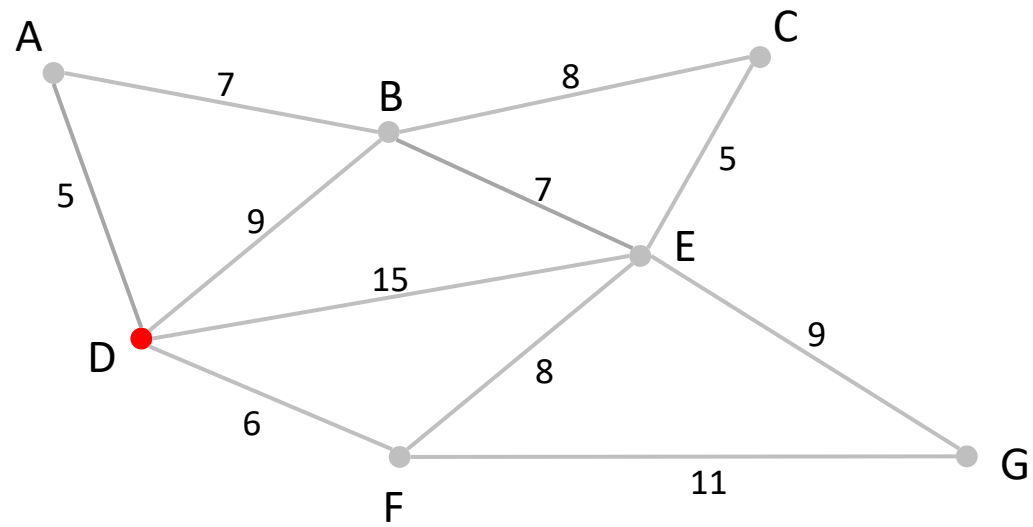
Prim algoritme



wel = []

niet = [A, B, C, D, E, F, G]

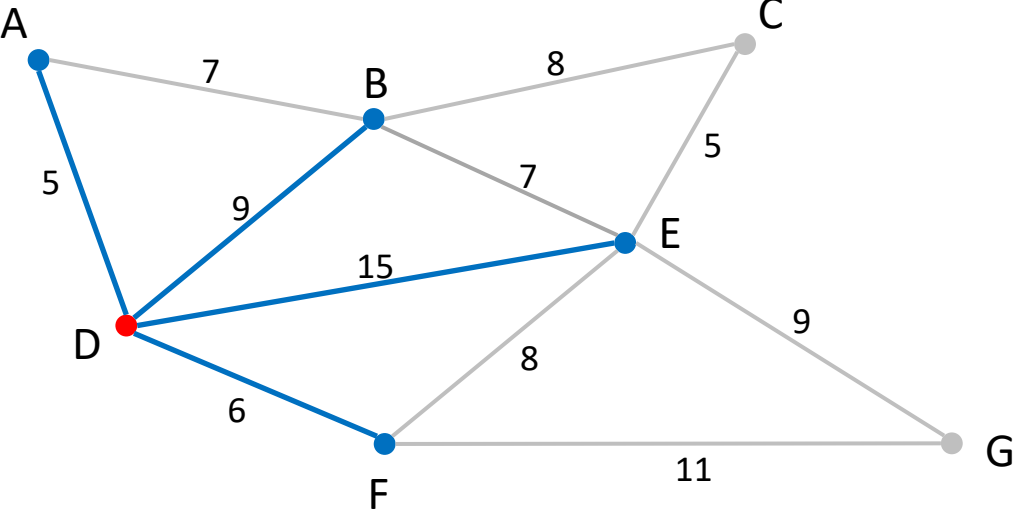
Prim algoritme



wel = [D]

niet = [A, B, C, E, F, G]

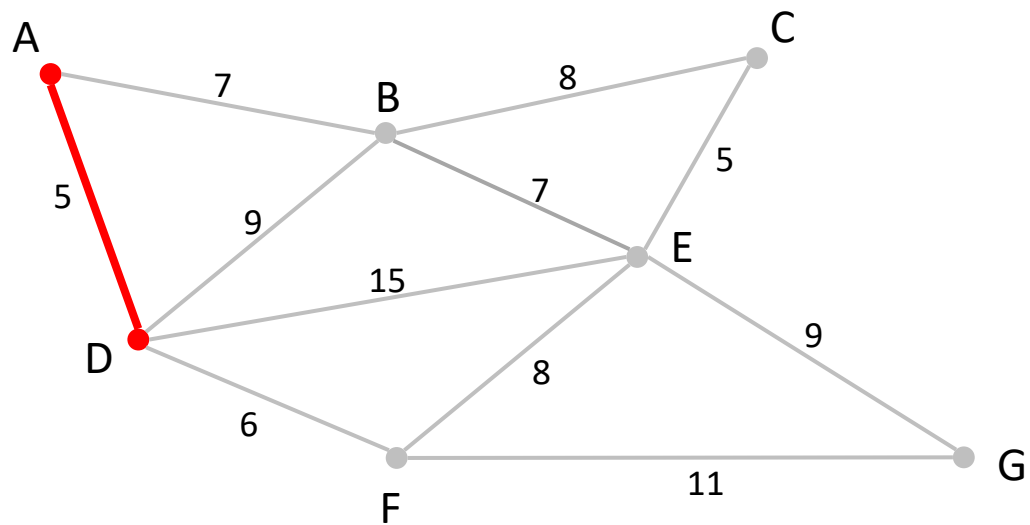
Prim algoritme



wel = [D]

niet = [A, B, C, E, F, G]

Prim algoritme

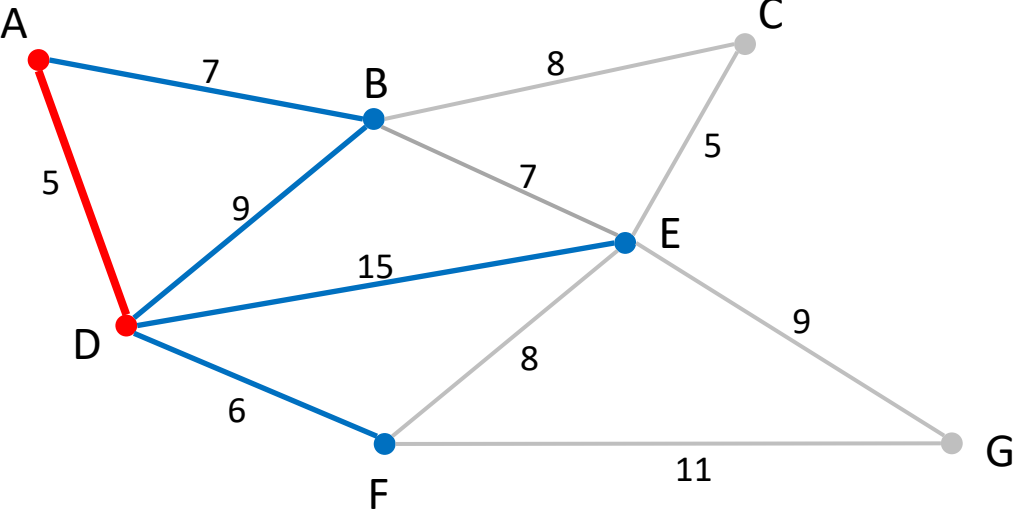


wel = [D, A]

DA

niet = [B, C, E, F, G]

Prim algoritme

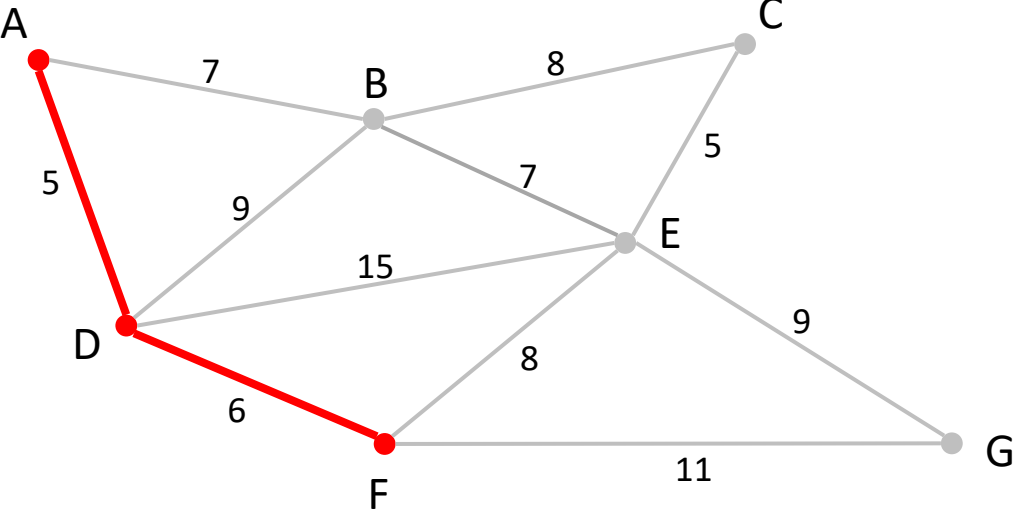


wel = [D, A]

DA

niet = [B, C, E, F, G]

Prim algoritme



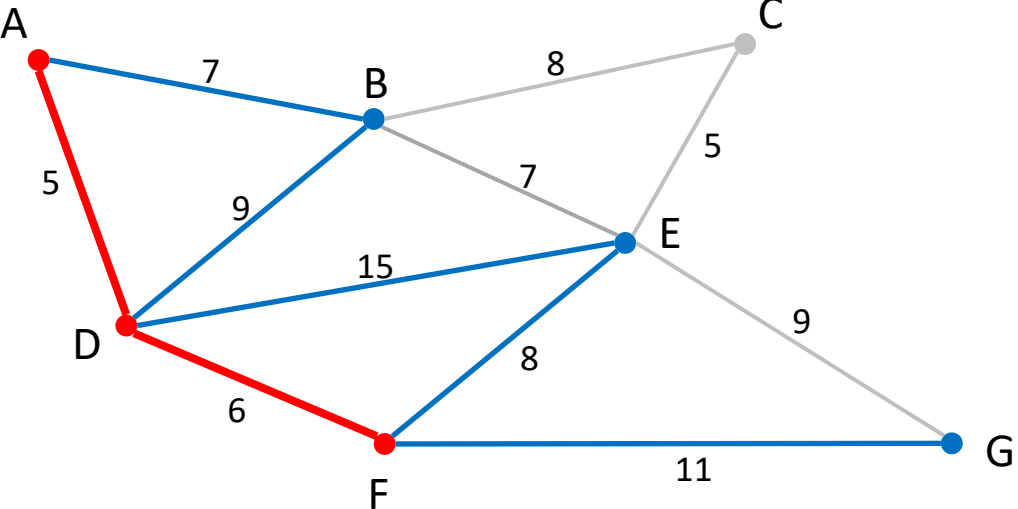
wel = [D, A, F]

niet = [B, C, E, G]

DA

DF

Prim algoritme



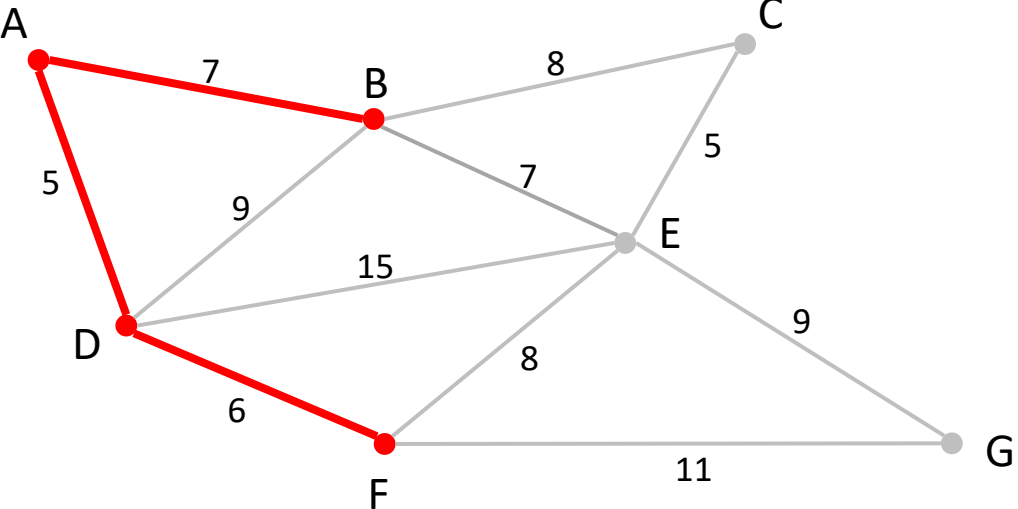
wel = [D, A, F]

niet = [B, C, E, G]

DA

DF

Prim algoritme

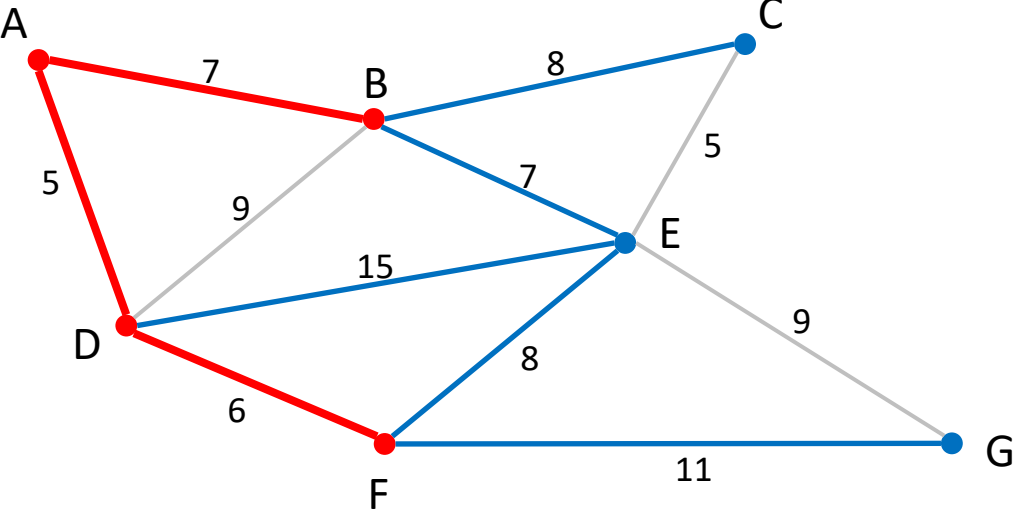


wel = [D, A, F, B]

niet = [C, E, G]

DA
DF
AB

Prim algoritme

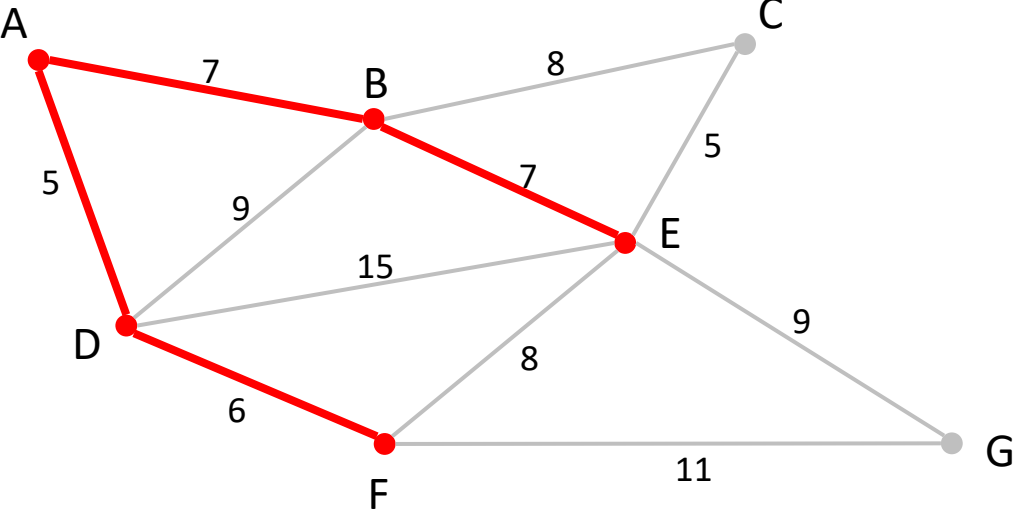


wel = [D, A, F, B]

niet = [C, E, G]

DA
DF
AB

Prim algoritme

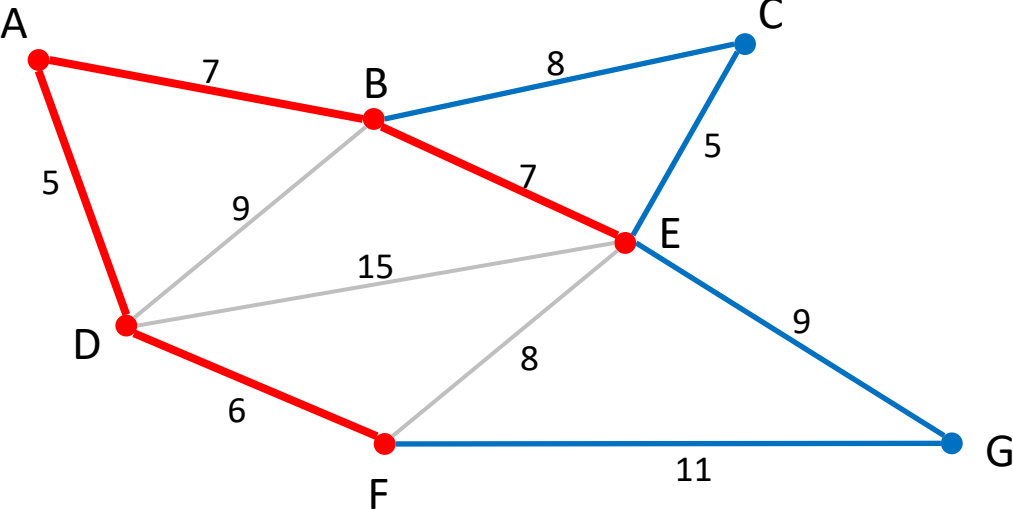


wel = [D, A, F, B, E]

niet = [C, G]

- DA
- DF
- AB
- BE

Prim algoritme

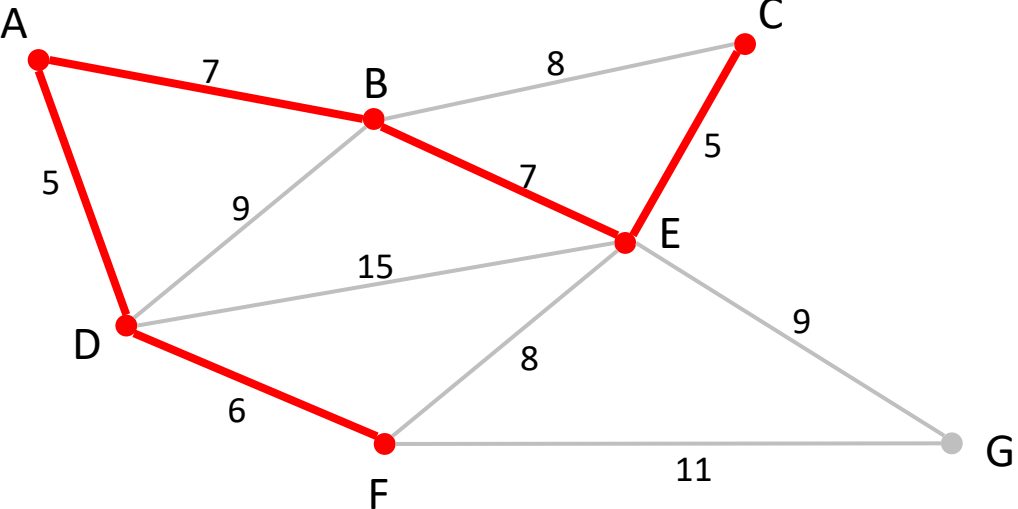


wel = [D, A, F, B, E]

niet = [C, G]

- DA
- DF
- AB
- BE

Prim algoritme

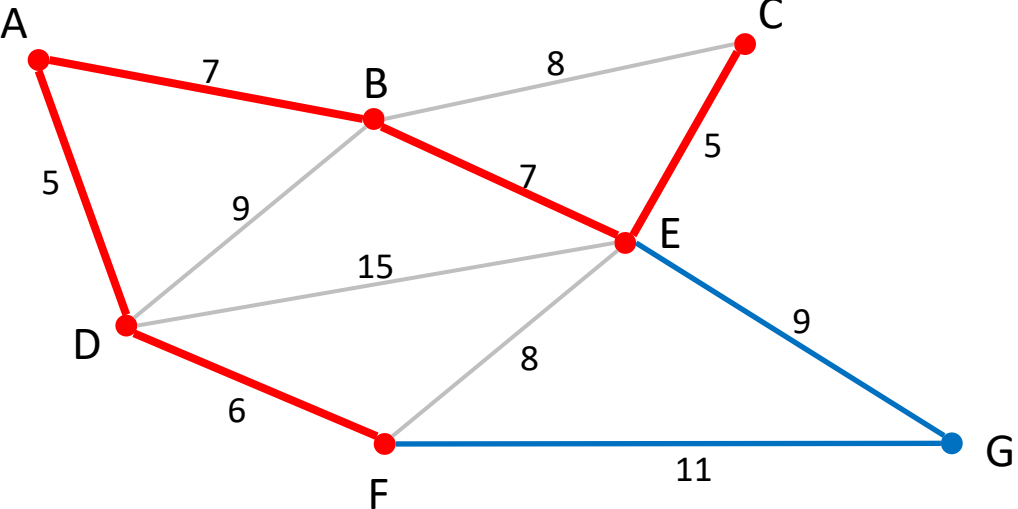


wel = [D, A, F, B, E, C]

niet = [G]

- DA
- DF
- AB
- BE
- EC

Prim algoritme

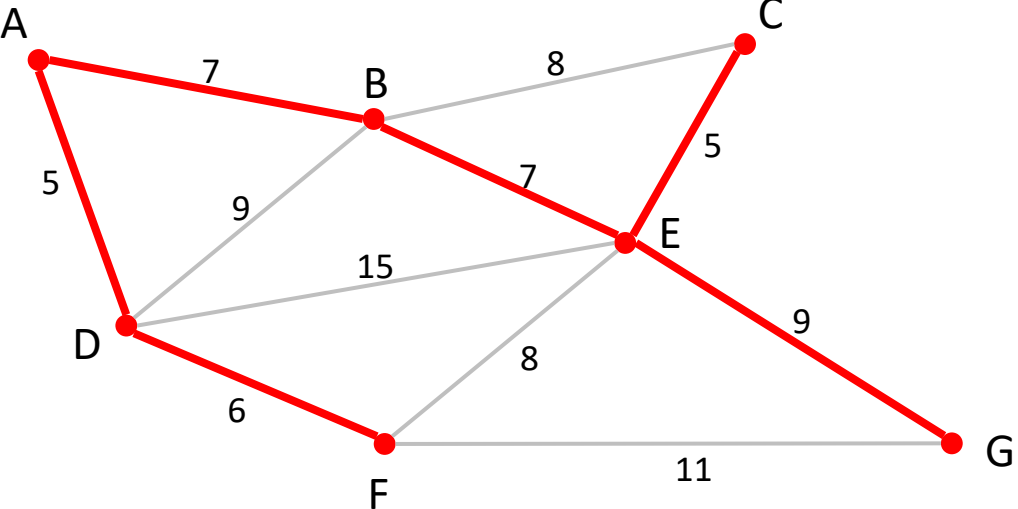


wel = [D, A, F, B, E, C]

niet = [G]

- DA
- DF
- AB
- BE
- EC

Prim algoritme



wel = [D, A, F, B, E, C, G]

niet = []

- DA
- DF
- AB
- BE
- EC
- EG

Prim algoritme

Terug

K-means algoritme



K-means algoritme

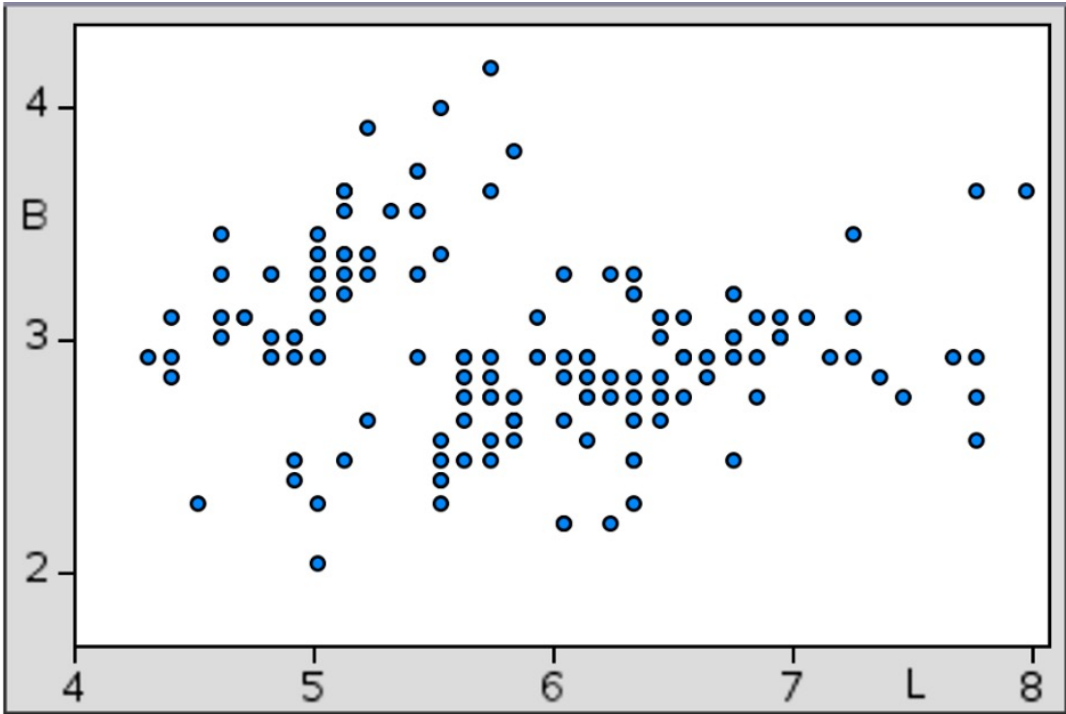


K-means algoritme



	A kl	B kb
=		
1	5.1	3.5
2	4.9	3.
3	4.7	3.2
4	4.6	3.1
5	5.	3.6
6	5.4	3.9
7	4.6	3.4
8	5.	3.4
9	4.4	2.9
10	4.9	3.1
11	5.4	3.7
12	4.8	3.4
13	4.8	3.
14	4.3	3.
15	5.8	4.
16	5.7	4.4

...



K-means algoritme

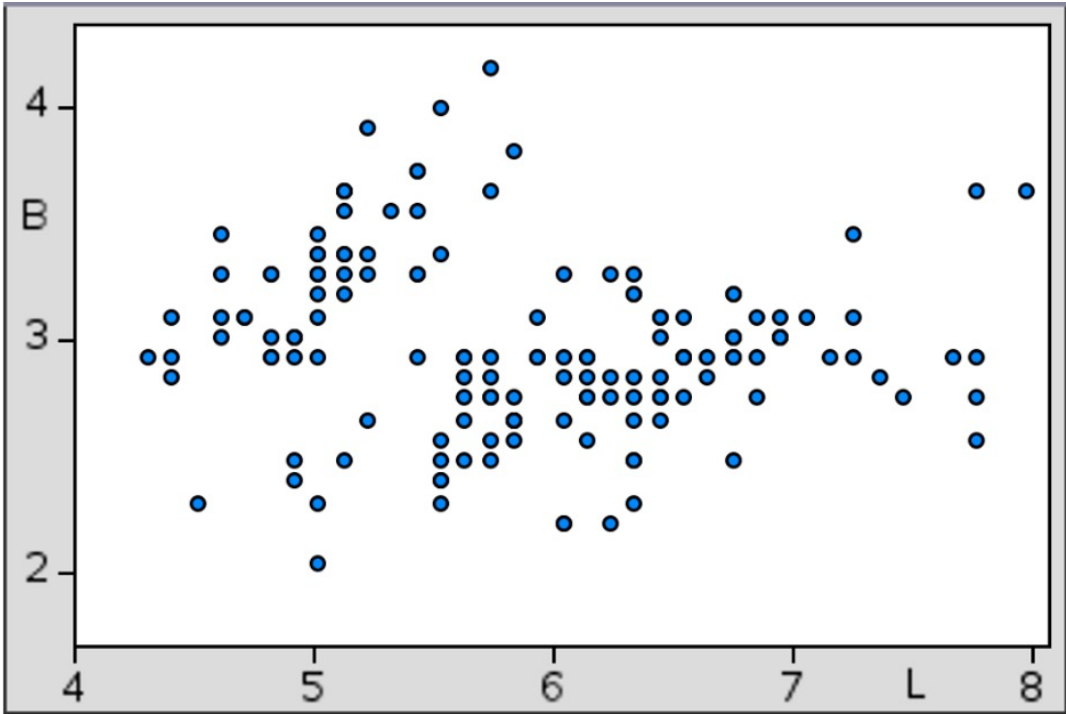
1. Kies willekeurige centra (elk met een eigen kleur).
2.
 - a. Bepaal voor elk punt uit de dataset het dichtstbijzijnde centrum en geef het punt dezelfde kleur als dat centrum.
 - b. Bereken voor elke (kleur)groep het gemiddelde (zwaartepunt) en neem dit als nieuw centrum.
3. Herhaal stap 2 net zolang totdat de centra niet meer van plaats veranderen.

K-means algoritme



	A kl	B kb
=		
1	5.1	3.5
2	4.9	3.
3	4.7	3.2
4	4.6	3.1
5	5.	3.6
6	5.4	3.9
7	4.6	3.4
8	5.	3.4
9	4.4	2.9
10	4.9	3.1
11	5.4	3.7
12	4.8	3.4
13	4.8	3.
14	4.3	3.
15	5.8	4.
16	5.7	4.4

...

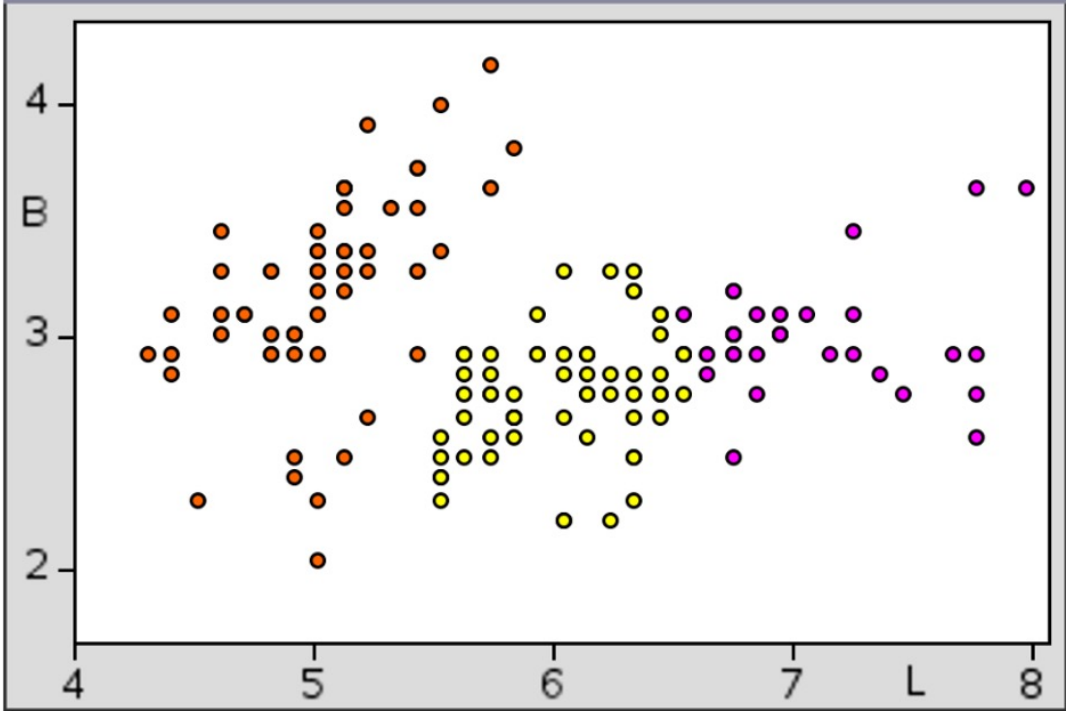


K-means algoritme



	A kl	B kb
=		
1	5.1	3.5
2	4.9	3.
3	4.7	3.2
4	4.6	3.1
5	5.	3.6
6	5.4	3.9
7	4.6	3.4
8	5.	3.4
9	4.4	2.9
10	4.9	3.1
11	5.4	3.7
12	4.8	3.4
13	4.8	3.
14	4.3	3.
15	5.8	4.
16	5.7	4.4

...



K-means algoritme

Terug

Torens van Hanoi



Torens van Hanoi

Terug

Langton's mier

Langtons mier is een twee dimensionale Cellulaire Automaat:

Een raster bestaande uit cellen die elk twee toestanden kunnen aannemen, 0 of 1.

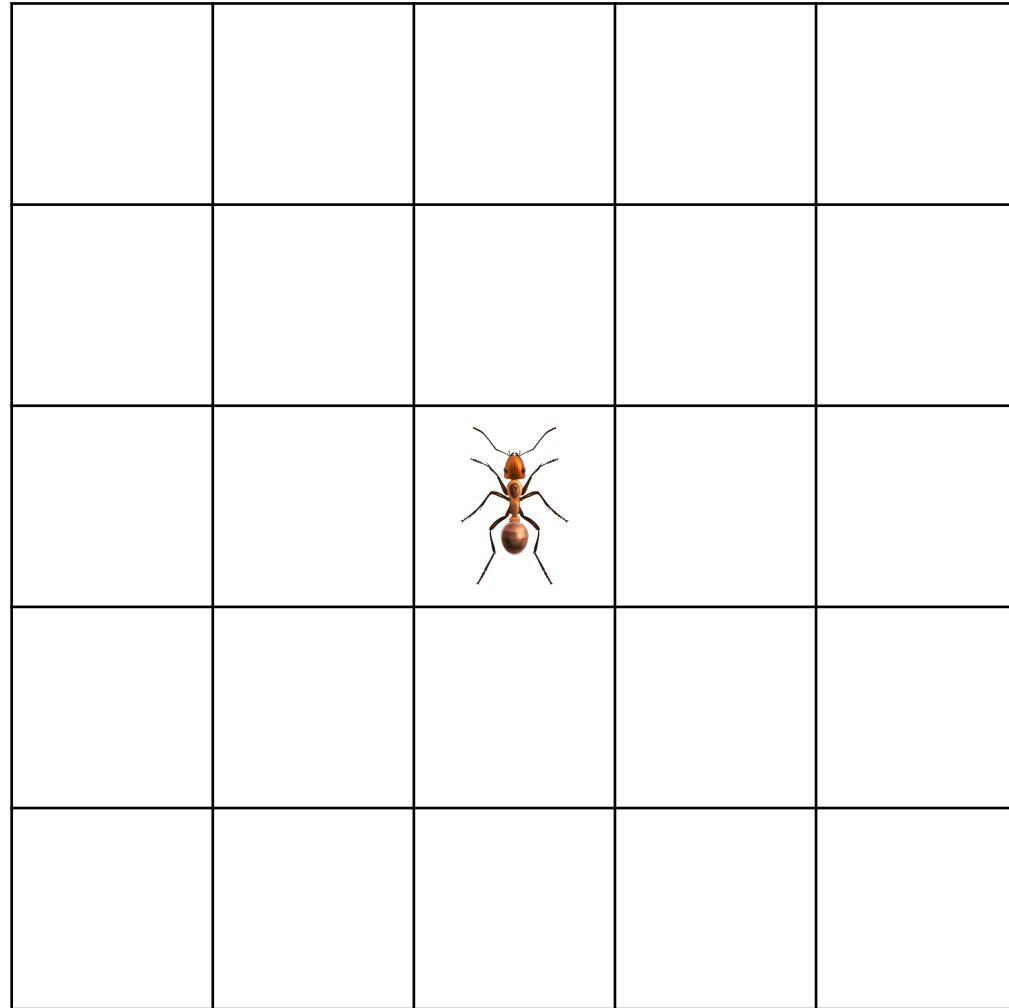
Elke volgende toestand wordt bepaald door de huidige toestand van alle cellen volgens vaste regels.

Langton's mier

Begin met een (oneindig groot) raster waarin alle cellen de waarde nul hebben.

Maak alle cellen met waarde 0 wit en maak alle cellen met waarde 1 zwart.

Plaats een mier in een van de cellen.



Langton's mier

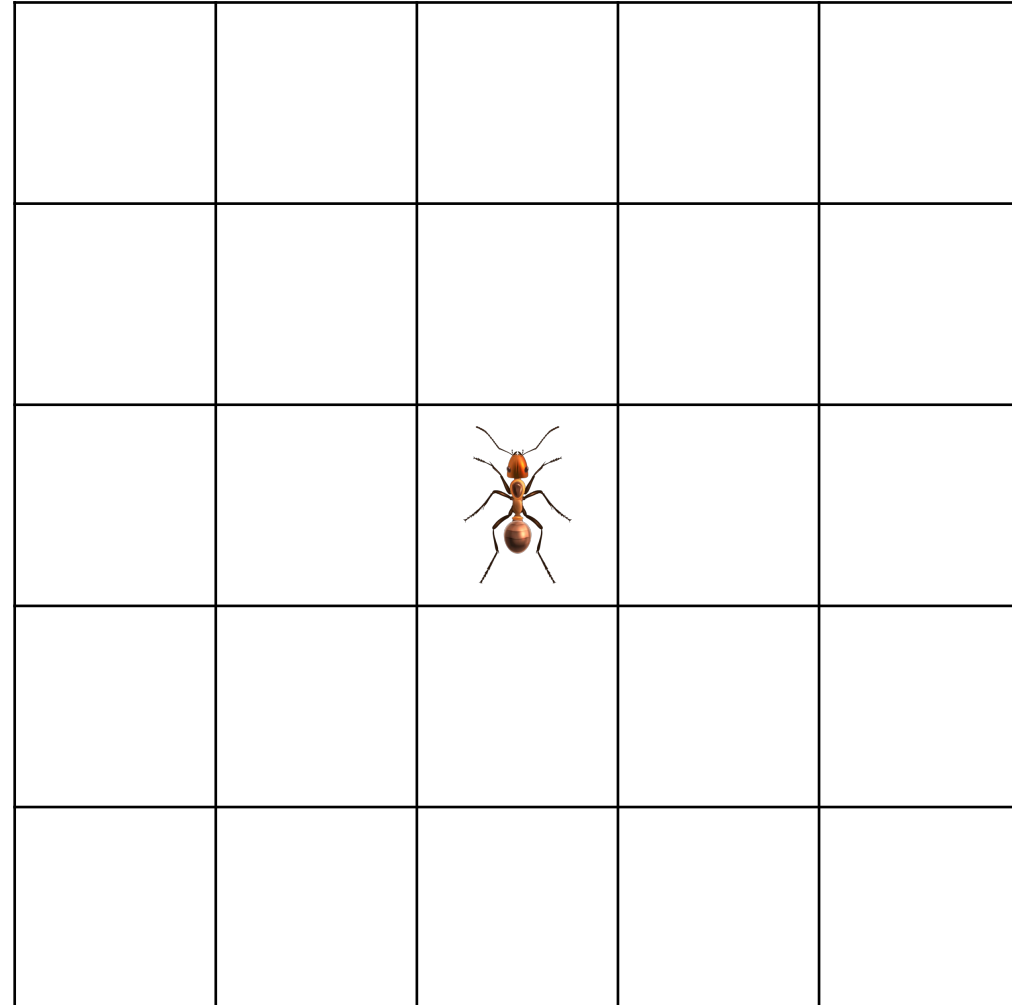
Regels:

Als de mier zich op een **wit** veld bevindt draai dan 90 graden naar **rechts**.

Doe een stap voorwaarts en verander de kleur van vorige cel.

Als de mier zich op een **gekleurd** veld bevindt draai dan 90 graden naar **links**.

Doe een stap voorwaarts en verander de kleur van vorige cel.



Langton's mier

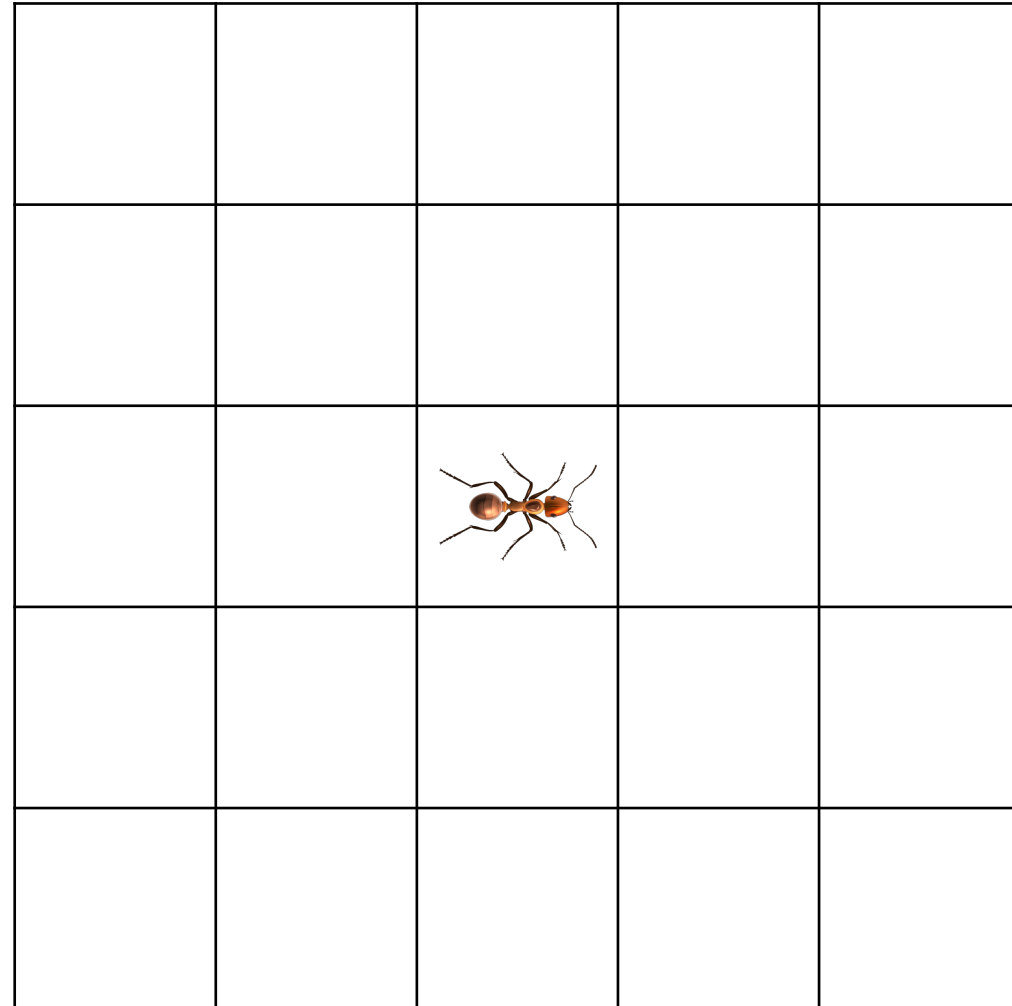
Regels:

Als de mier zich op een **wit** veld bevindt draai dan 90 graden naar **rechts**.

Doe een stap voorwaarts en verander de kleur van vorige cel.

Als de mier zich op een **gekleurd** veld bevindt draai dan 90 graden naar **links**.

Doe een stap voorwaarts en verander de kleur van vorige cel.



Langton's mier

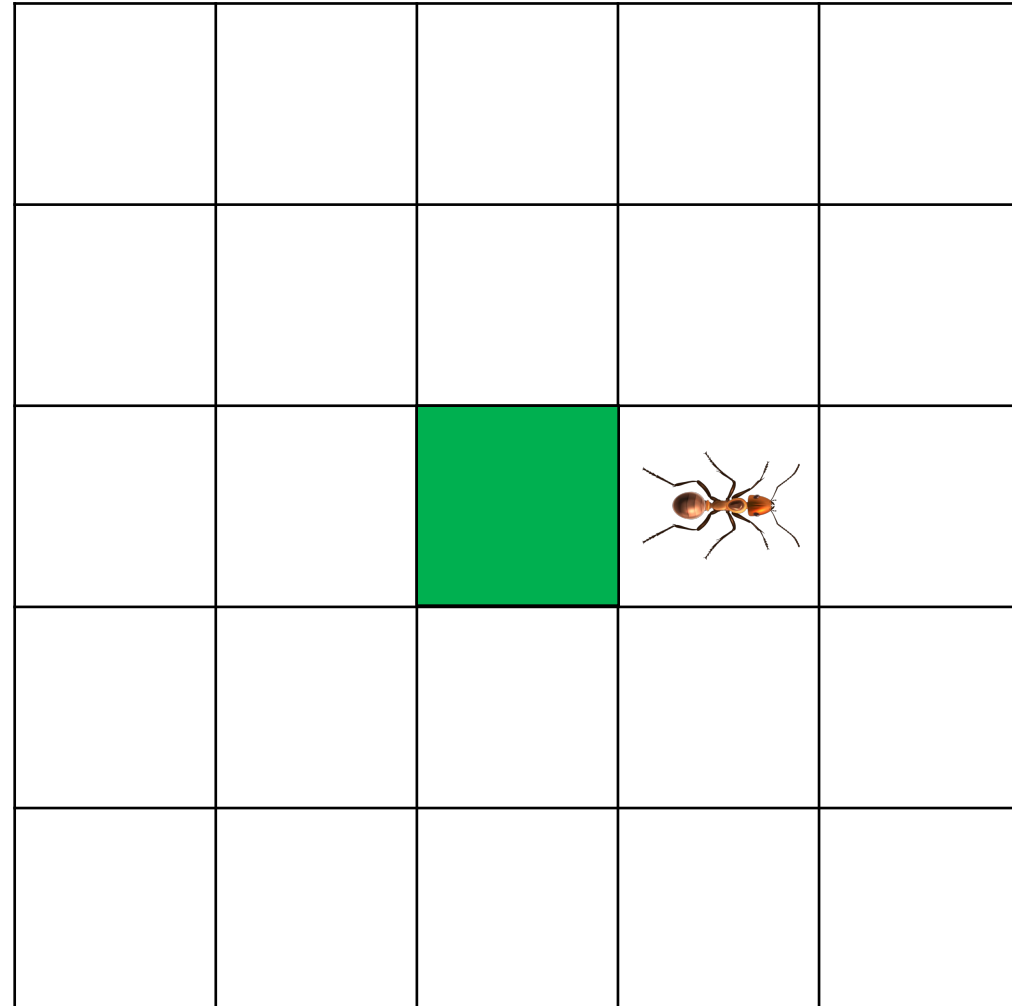
Regels:

Als de mier zich op een **wit** veld bevindt draai dan 90 graden naar **rechts**.

Doe een stap voorwaarts en verander de kleur van vorige cel.

Als de mier zich op een **gekleurd** veld bevindt draai dan 90 graden naar **links**.

Doe een stap voorwaarts en verander de kleur van vorige cel.



Langton's mier

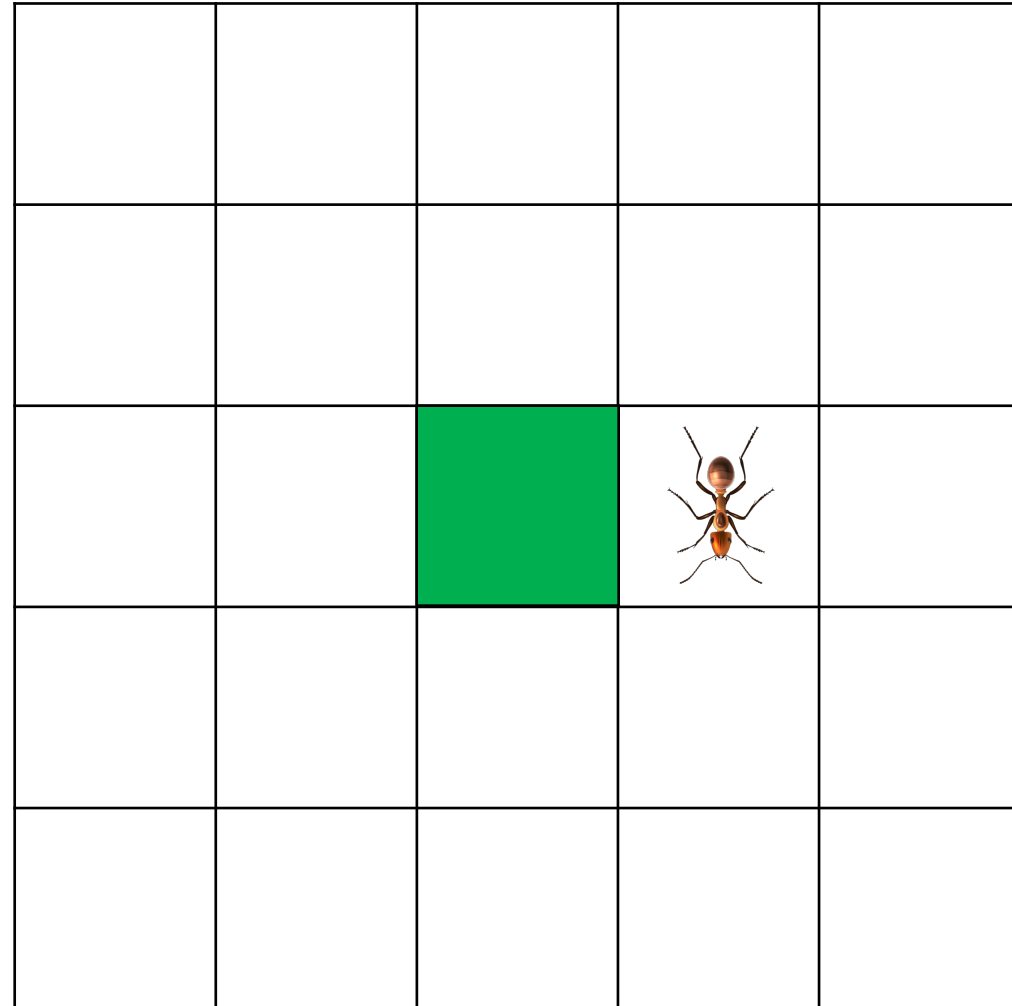
Regels:

Als de mier zich op een **wit** veld bevindt draai dan 90 graden naar **rechts**.

Doe een stap voorwaarts en verander de kleur van vorige cel.

Als de mier zich op een **gekleurd** veld bevindt draai dan 90 graden naar **links**.

Doe een stap voorwaarts en verander de kleur van vorige cel.



Langton's mier

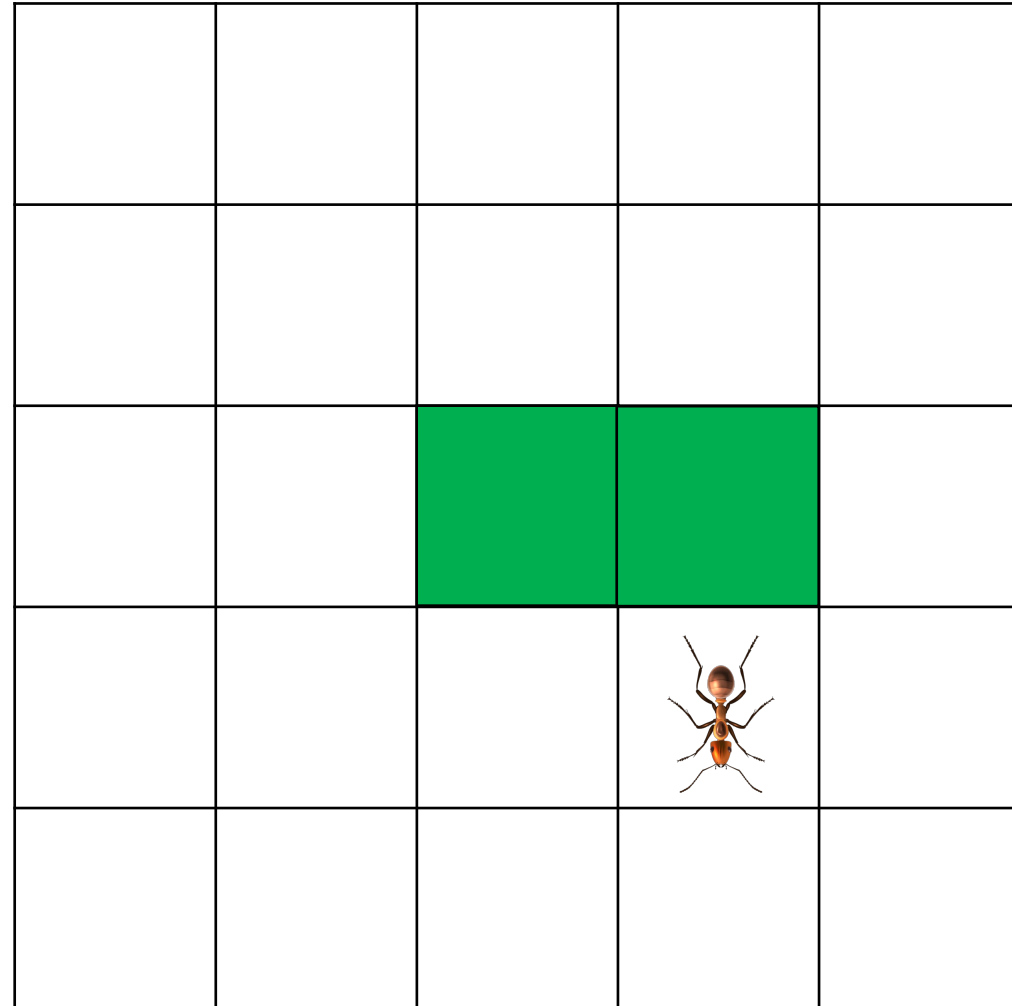
Regels:

Als de mier zich op een **wit** veld bevindt draai dan 90 graden naar **rechts**.

Doe een stap voorwaarts en verander de kleur van vorige cel.

Als de mier zich op een **gekleurd** veld bevindt draai dan 90 graden naar **links**.

Doe een stap voorwaarts en verander de kleur van vorige cel.



Langton's mier

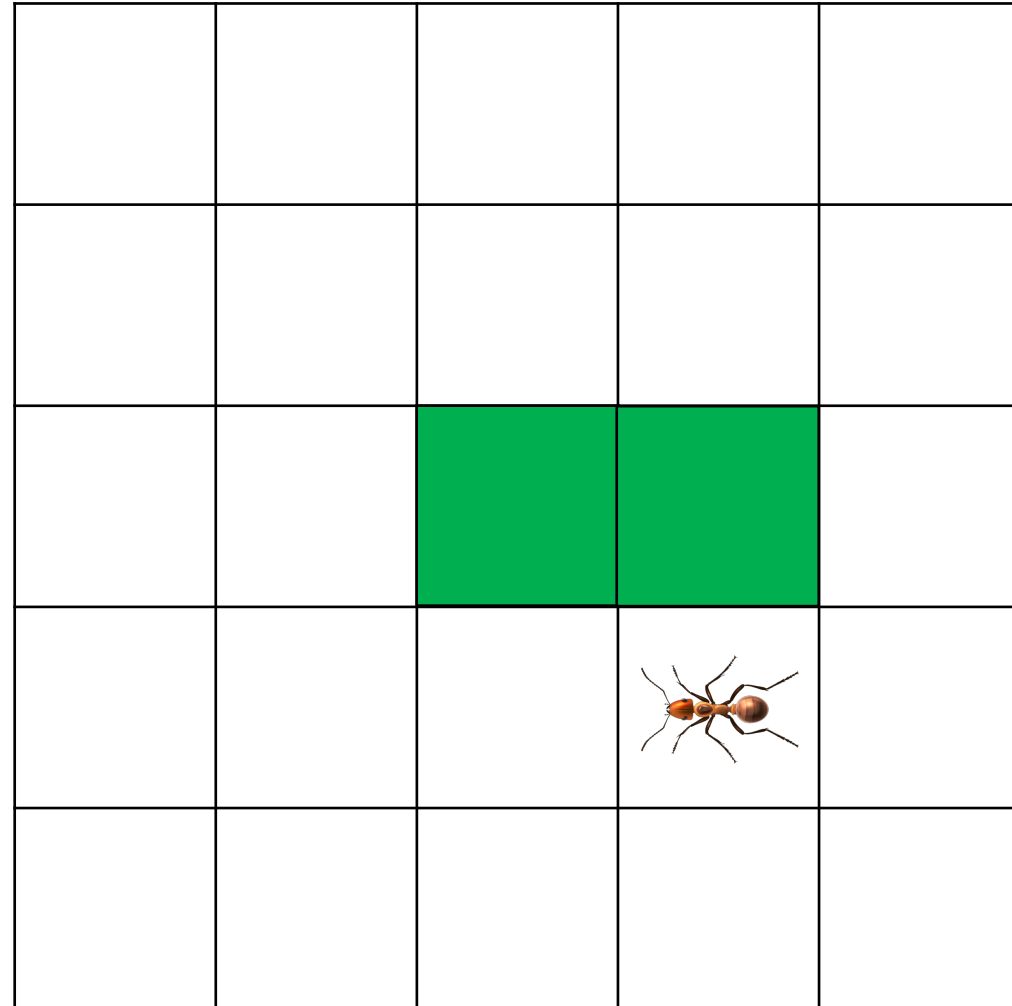
Regels:

Als de mier zich op een **wit** veld bevindt draai dan 90 graden naar **rechts**.

Doe een stap voorwaarts en verander de kleur van vorige cel.

Als de mier zich op een **gekleurd** veld bevindt draai dan 90 graden naar **links**.

Doe een stap voorwaarts en verander de kleur van vorige cel.



Langton's mier

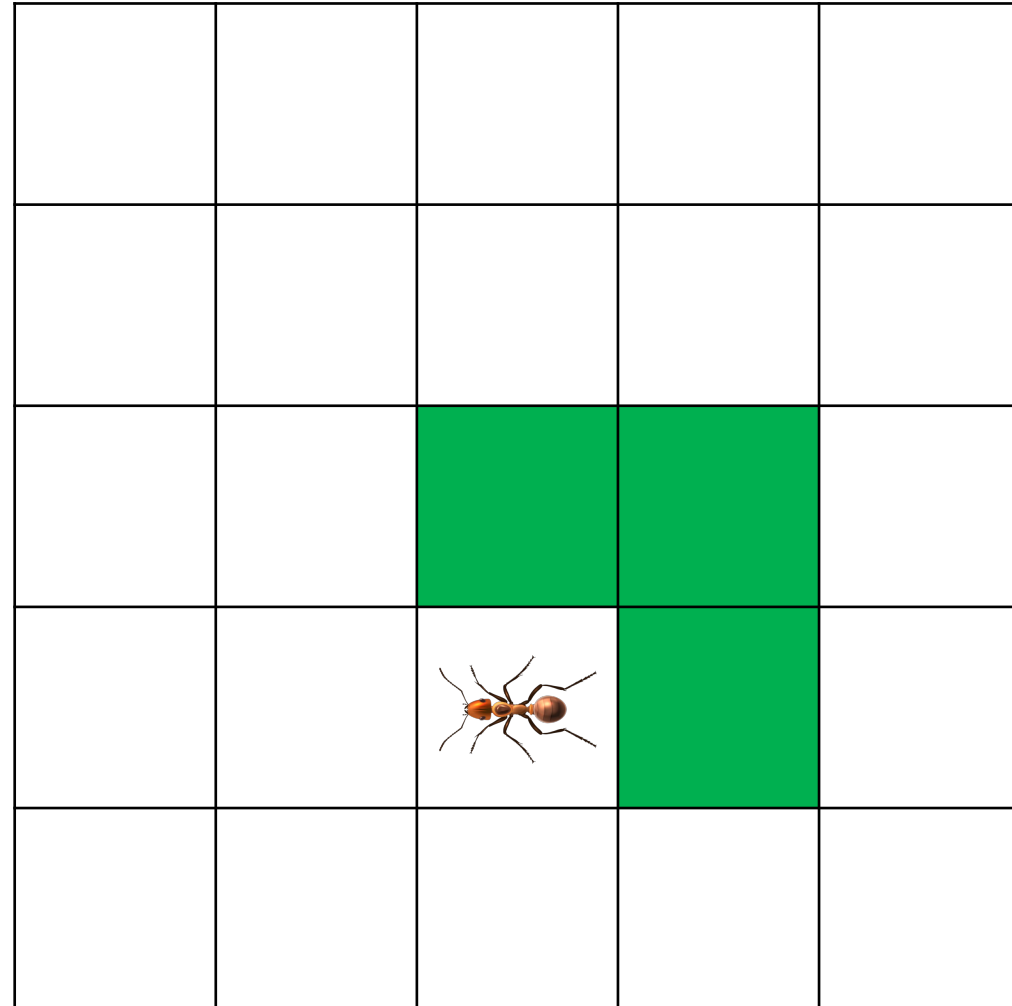
Regels:

Als de mier zich op een **wit** veld bevindt draai dan 90 graden naar **rechts**.

Doe een stap voorwaarts en verander de kleur van vorige cel.

Als de mier zich op een **gekleurd** veld bevindt draai dan 90 graden naar **links**.

Doe een stap voorwaarts en verander de kleur van vorige cel.



Langton's mier

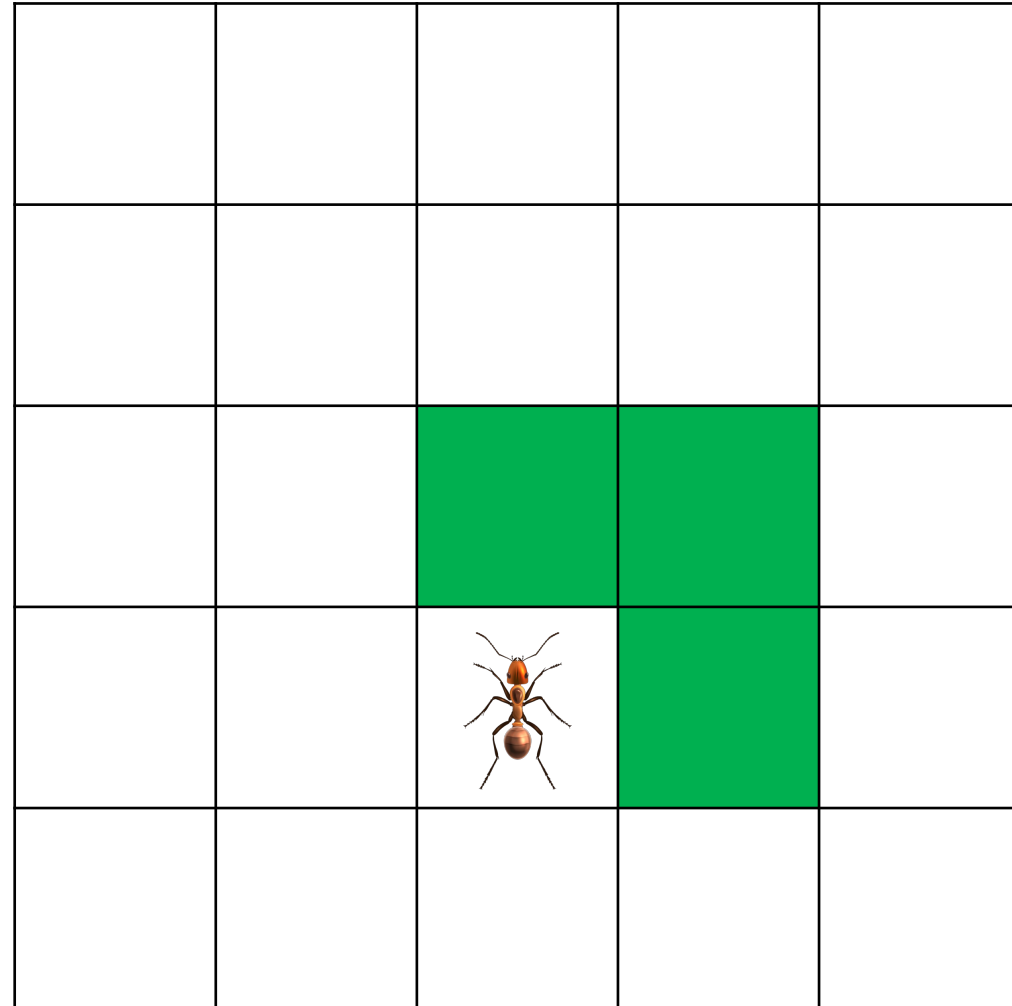
Regels:

Als de mier zich op een **wit** veld bevindt draai dan 90 graden naar **rechts**.

Doe een stap voorwaarts en verander de kleur van vorige cel.

Als de mier zich op een **gekleurd** veld bevindt draai dan 90 graden naar **links**.

Doe een stap voorwaarts en verander de kleur van vorige cel.



Langton's mier

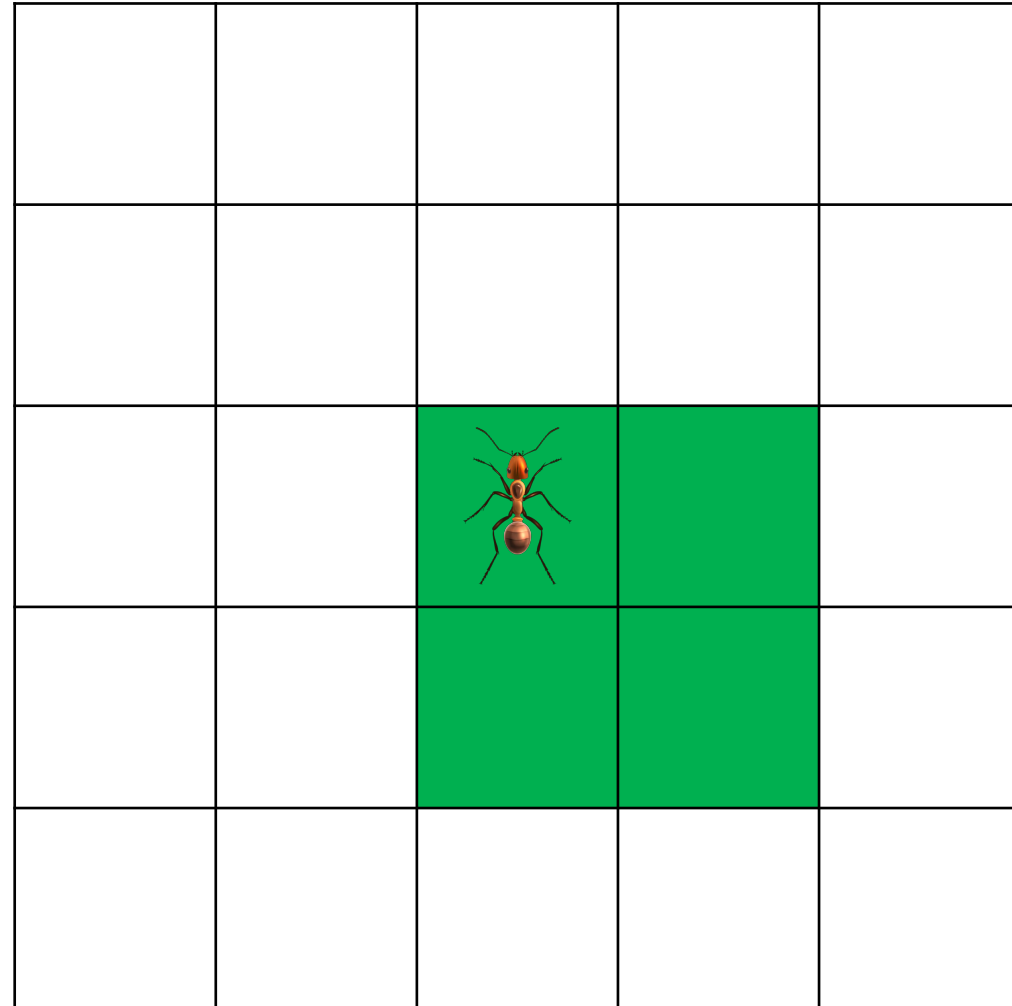
Regels:

Als de mier zich op een **wit** veld bevindt draai dan 90 graden naar **rechts**.

Doe een stap voorwaarts en verander de kleur van vorige cel.

Als de mier zich op een **gekleurd** veld bevindt draai dan 90 graden naar **links**.

Doe een stap voorwaarts en verander de kleur van vorige cel.



Langton's mier

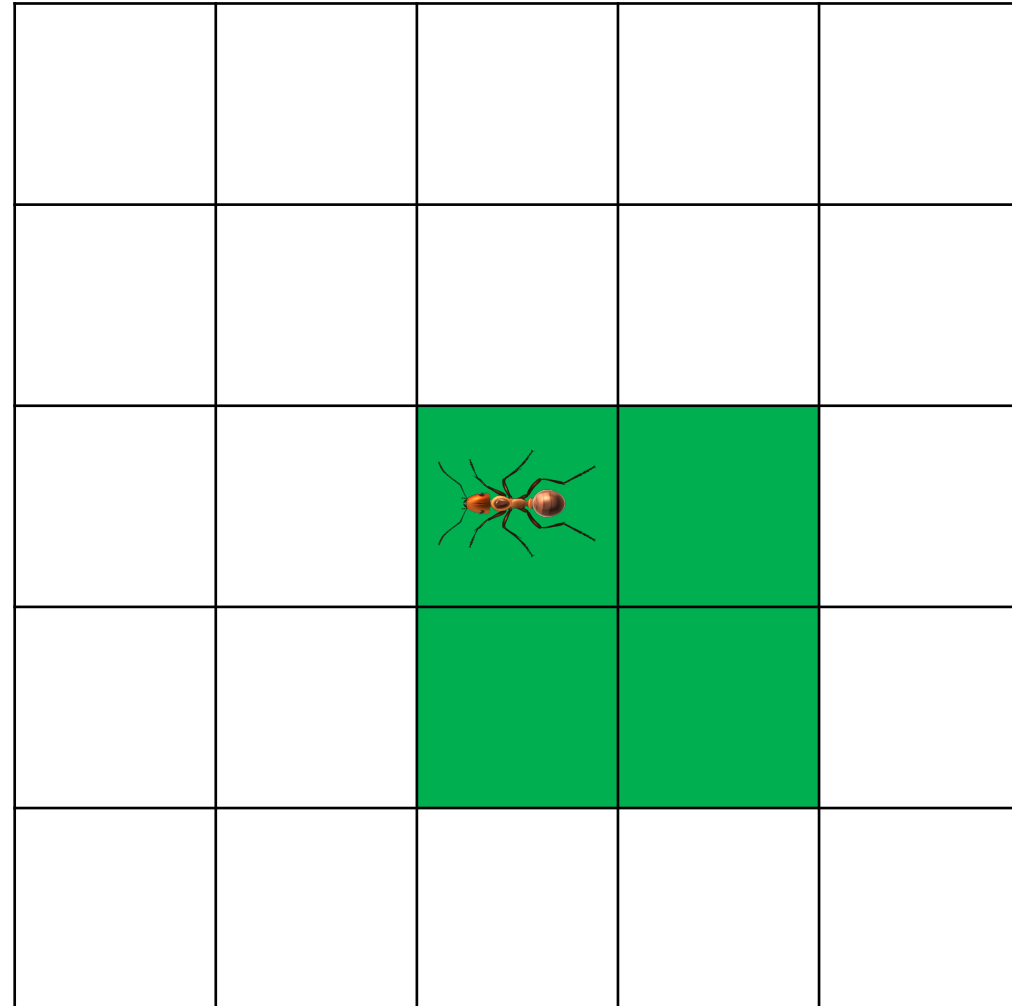
Regels:

Als de mier zich op een **wit** veld bevindt draai dan 90 graden naar **rechts**.

Doe een stap voorwaarts en verander de kleur van vorige cel.

Als de mier zich op een **gekleurd** veld bevindt draai dan 90 graden naar **links**.

Doe een stap voorwaarts en verander de kleur van vorige cel.



Langton's mier

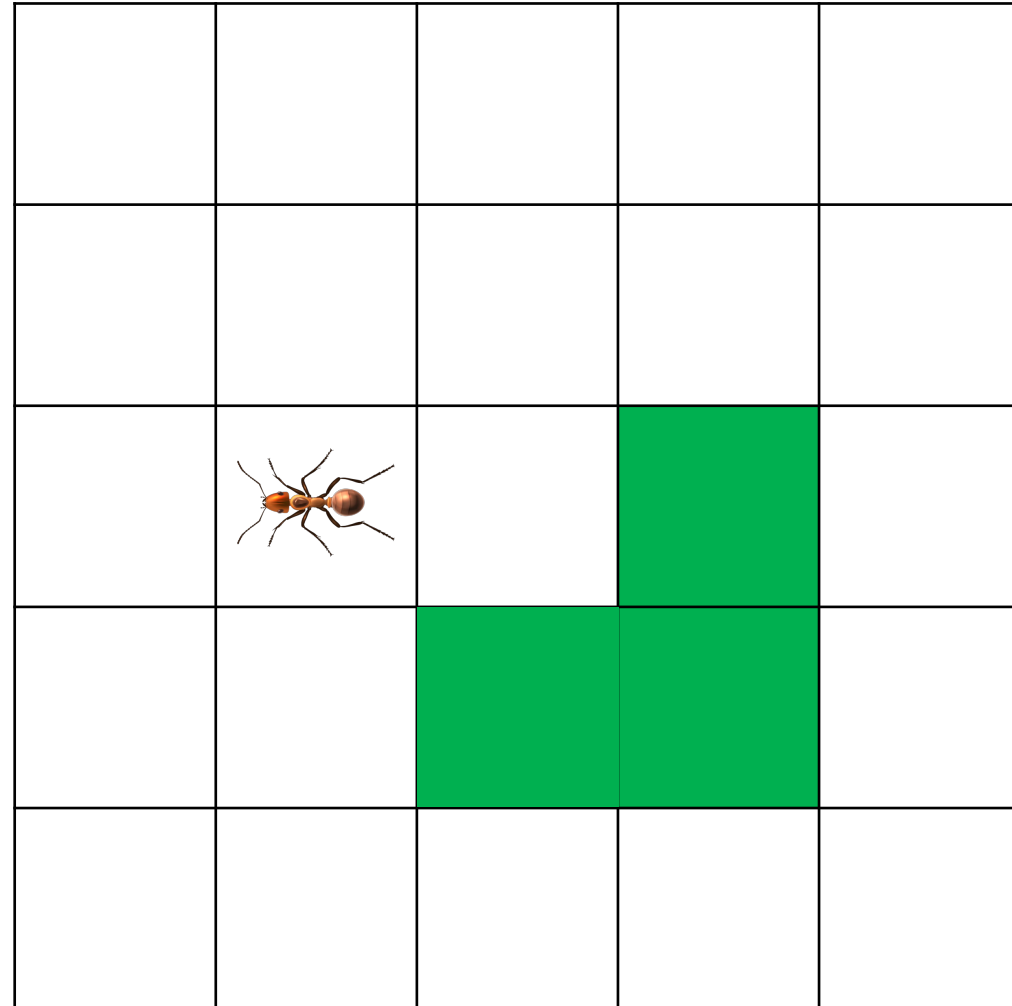
Regels:

Als de mier zich op een **wit** veld bevindt draai dan 90 graden naar **rechts**.

Doe een stap voorwaarts en verander de kleur van vorige cel.

Als de mier zich op een **gekleurd** veld bevindt draai dan 90 graden naar **links**.

Doe een stap voorwaarts en verander de kleur van vorige cel.



Langton's mier

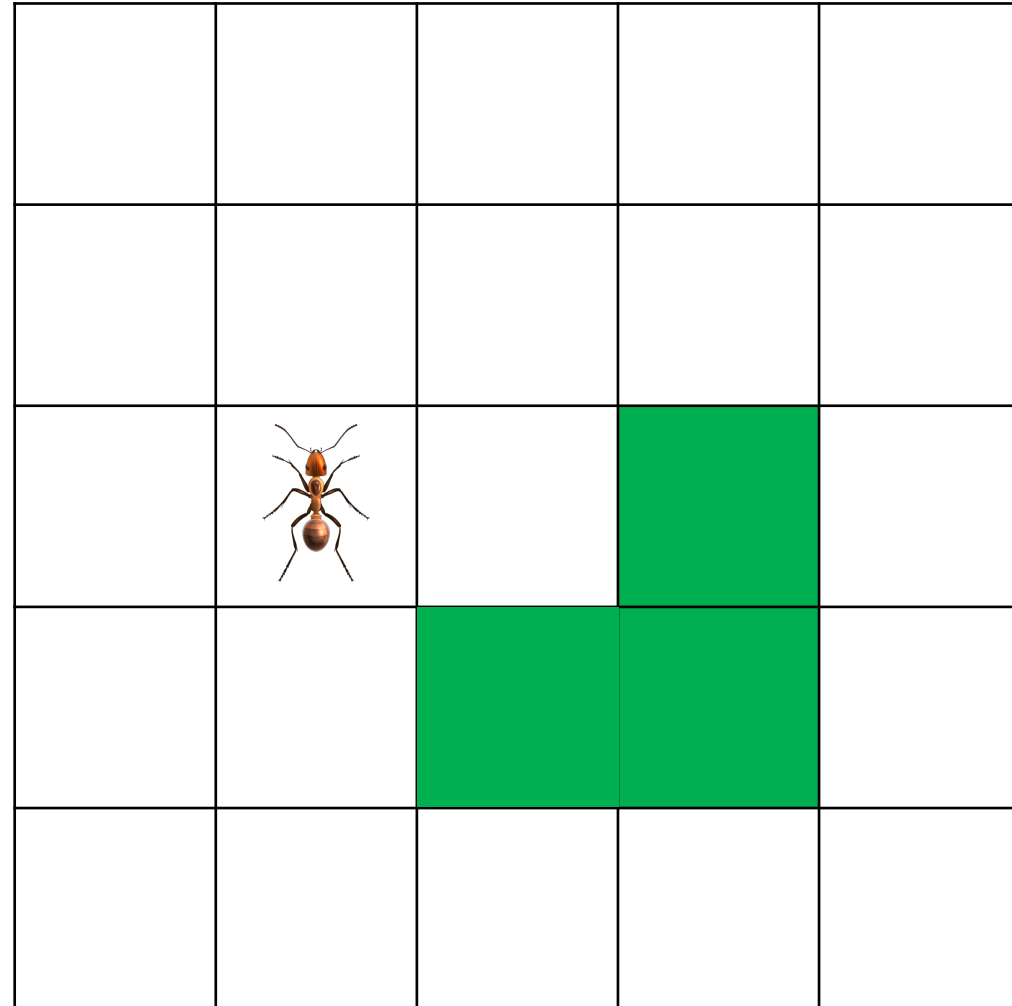
Regels:

Als de mier zich op een **wit** veld bevindt draai dan 90 graden naar **rechts**.

Doe een stap voorwaarts en verander de kleur van vorige cel.

Als de mier zich op een **gekleurd** veld bevindt draai dan 90 graden naar **links**.

Doe een stap voorwaarts en verander de kleur van vorige cel.



Langton's mier

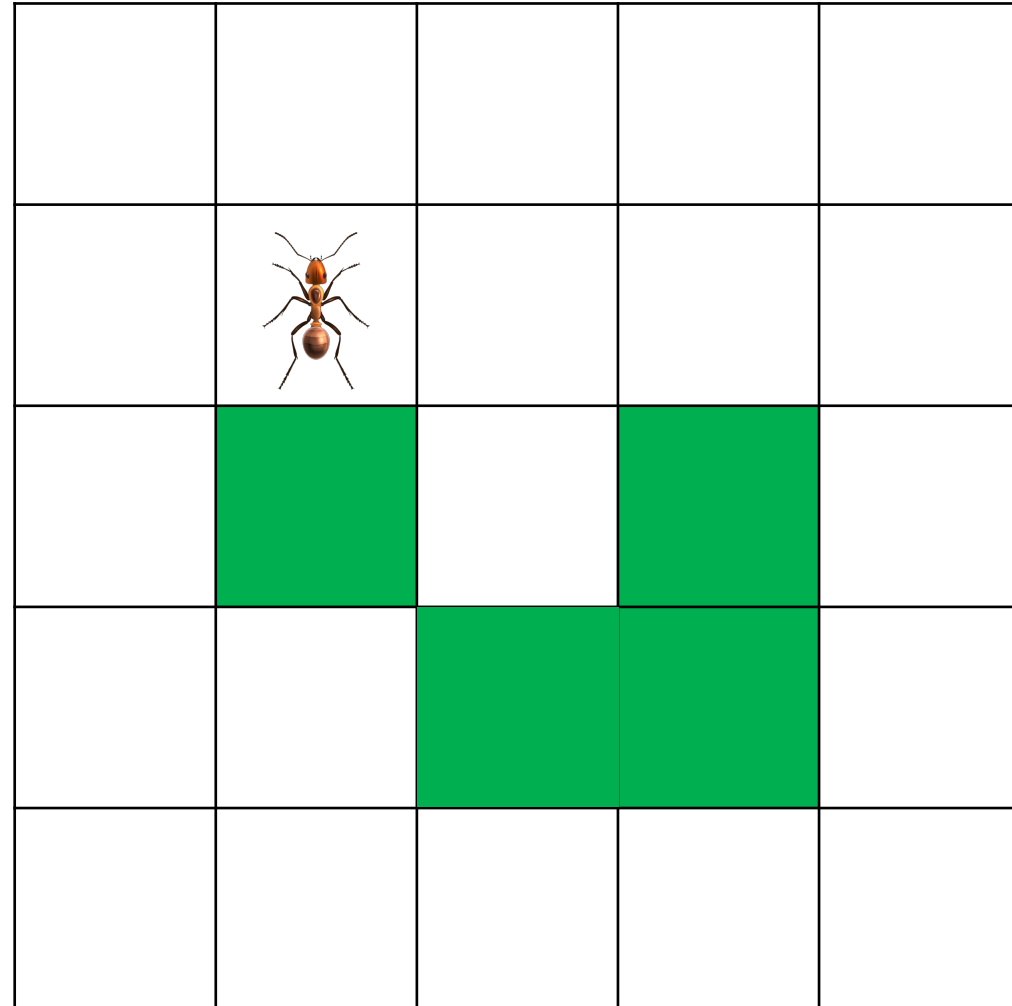
Regels:

Als de mier zich op een **wit** veld bevindt draai dan 90 graden naar **rechts**.

Doe een stap voorwaarts en verander de kleur van vorige cel.

Als de mier zich op een **gekleurd** veld bevindt draai dan 90 graden naar **links**.

Doe een stap voorwaarts en verander de kleur van vorige cel.



Langton's mier

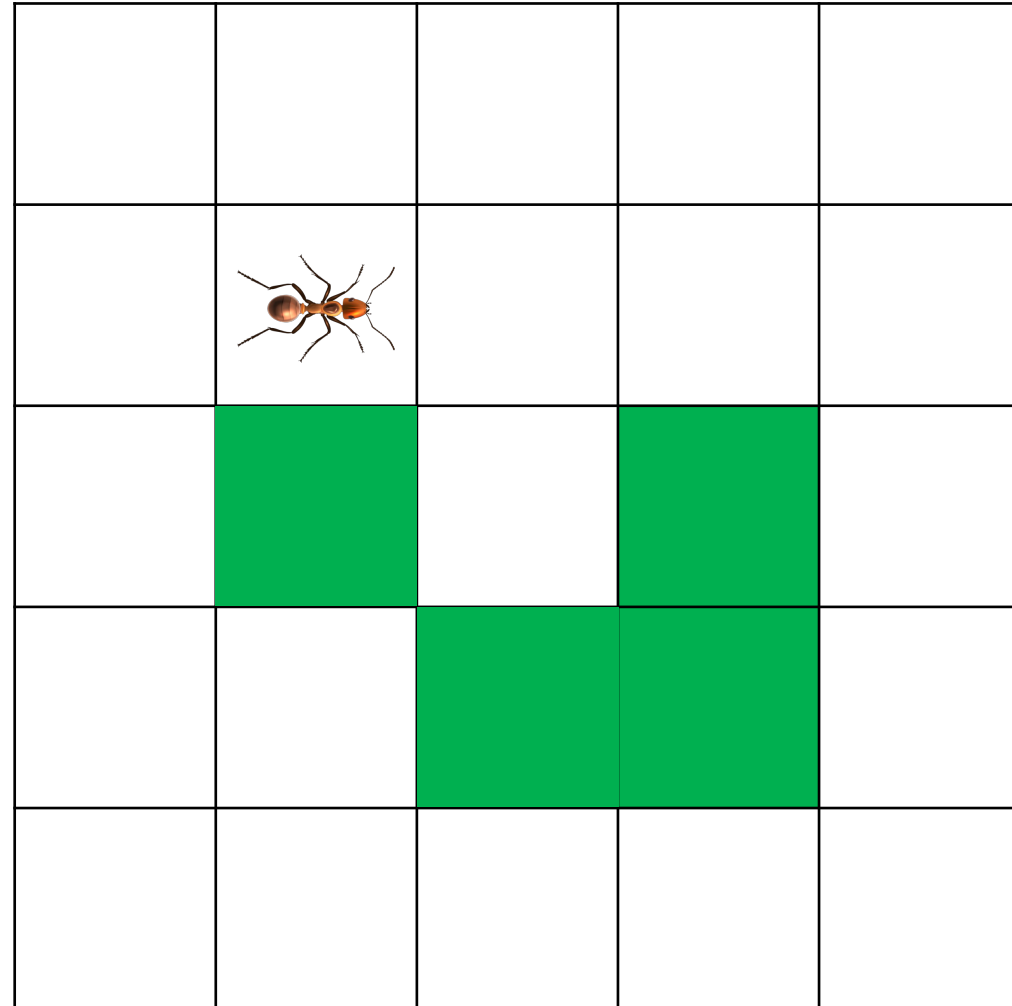
Regels:

Als de mier zich op een **wit** veld bevindt draai dan 90 graden naar **rechts**.

Doe een stap voorwaarts en verander de kleur van vorige cel.

Als de mier zich op een **gekleurd** veld bevindt draai dan 90 graden naar **links**.

Doe een stap voorwaarts en verander de kleur van vorige cel.



Langton's mier

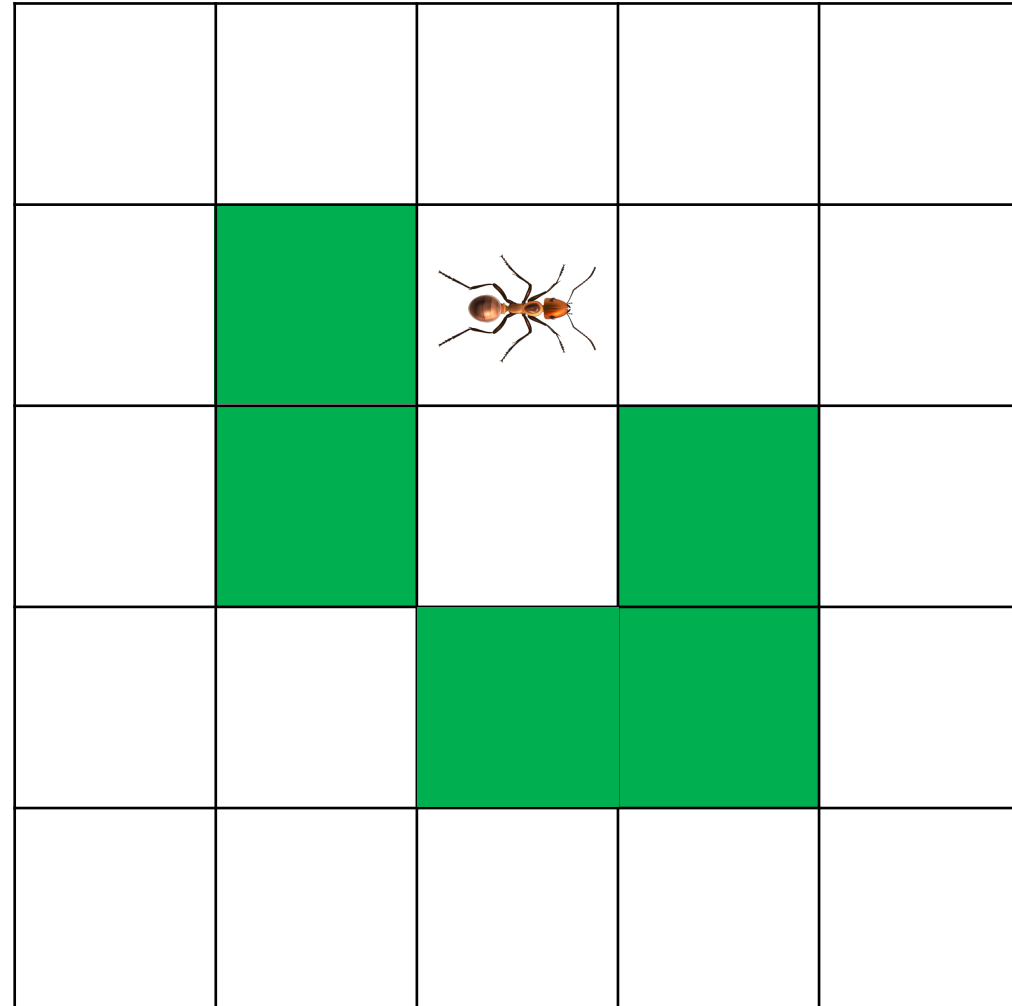
Regels:

Als de mier zich op een **wit** veld bevindt draai dan 90 graden naar **rechts**.

Doe een stap voorwaarts en verander de kleur van vorige cel.

Als de mier zich op een **gekleurd** veld bevindt draai dan 90 graden naar **links**.

Doe een stap voorwaarts en verander de kleur van vorige cel.



Langton's mier

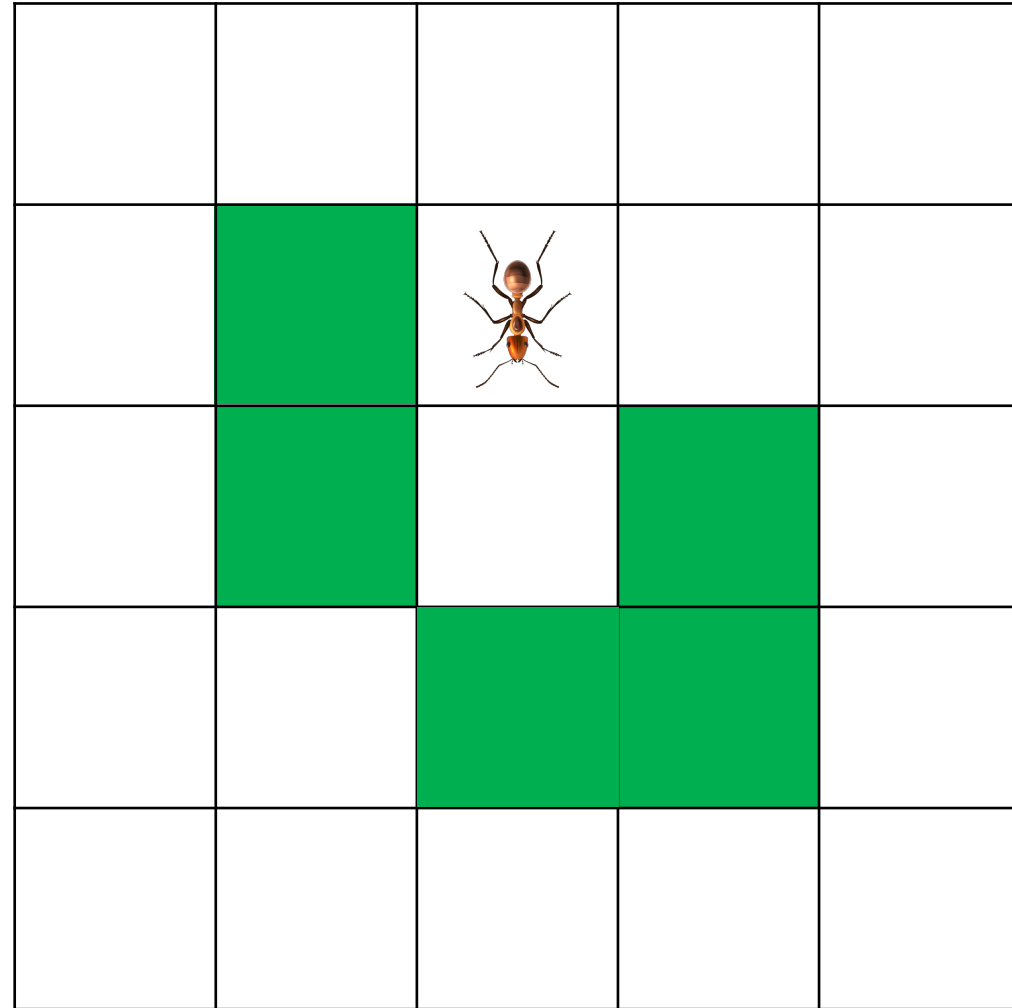
Regels:

Als de mier zich op een **wit** veld bevindt draai dan 90 graden naar **rechts**.

Doe een stap voorwaarts en verander de kleur van vorige cel.

Als de mier zich op een **gekleurd** veld bevindt draai dan 90 graden naar **links**.

Doe een stap voorwaarts en verander de kleur van vorige cel.



Langton's mier

Terug