

# Vorbereitung auf die 2. Teilprüfung

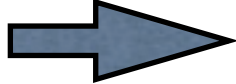
# Stoff der 2. Teilprüfung

- Algorithmus entwerfen, implementieren
- Collections (ArrayList, Stack, Maps)
- Containertypen: String, Arrays
- Interfaces
- Rekursion

# Beispiele

- Aufgabe 1: Algorithmus, String
- Aufgabe 2: Klassen und Interfaces schreiben
- Aufgabe 1/2: Variante mit Rekursion

# Aufgabe I

- Umgang mit Strings
- Lauflängenkodierung (Runlength Encoding)
- „aaaFFgGGGabbbbb“  „3a2Fg3Ga5b“

# Lauf­län­gen­ko­die­rung

internal

a	a	a	F	F	g	G	G	G	a	b	b	b	b	b
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↑    ↑  
start current

repeat = 0

compressed= null

# Lauf­län­gen­ko­die­rung

internal

a	a	a	F	F	g	G	G	G	a	b	b	b	b	b
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↑    ↑  
start current

repeat = 1

compressed= null

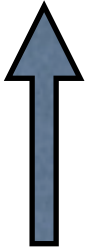
# Lauf­län­gen­ko­die­rung

internal

a	a	a	F	F	g	G	G	G	a	b	b	b	b	b
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



start



current

repeat = 2

compressed= null

# Lauf­län­gen­ko­die­rung

internal



↑  
start

↑  
current

repeat = 2

compressed = 

3	a
---	---

# Lauf­län­gen­ko­die­rung

internal

a	a	a	F	F	g	G	G	G	a	b	b	b	b	b
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



start



current

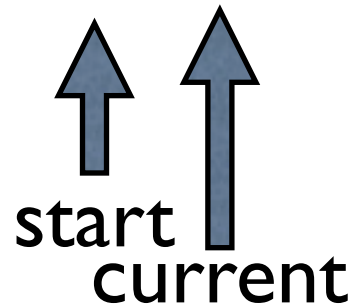
repeat = 0

compressed= 

3	a
---	---

# Lauf­län­gen­ko­die­rung

internal



repeat = 1

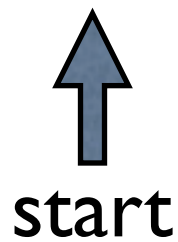
compressed = 

3	a
---	---

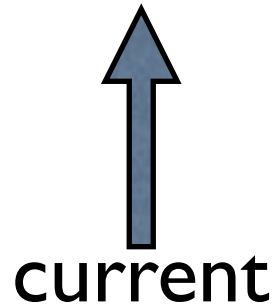
# Lauf­län­gen­ko­die­rung

internal

a	a	a	F	F	g	G	G	G	a	b	b	b	b	b
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



start



current

repeat = 1

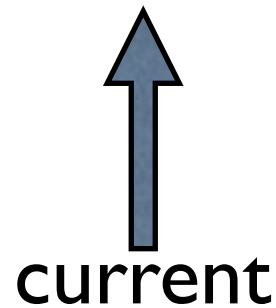
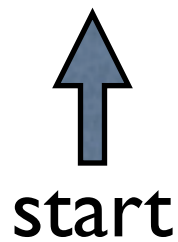
compressed = 

3	a	2	F
---	---	---	---

# Lauf­län­gen­ko­die­rung

internal

a	a	a	F	F	g	G	G	G	a	b	b	b	b	b
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---



repeat = 0

compressed=

3	a	2	F
---	---	---	---

# Lauf­län­gen­ko­die­rung

internal

a	a	a	F	F	g	G	G	G	a	b	b	b	b	b
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↑      ↑  
start   current

repeat = 0

compressed= 

3	a	2	F	g
---	---	---	---	---

# Laufänglenkodierung

internal

a	a	a	F	F	g	G	G	G	a	b	b	b	b	b
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↑    ↑  
start current

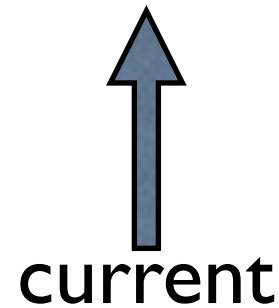
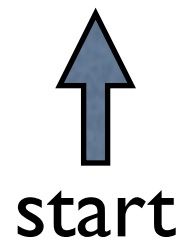
repeat = 1

compressed= 

3	a	2	F	g
---	---	---	---	---

# Laufängenkodierung

internal



repeat = 2

compressed=

3	a	2	F	g
---	---	---	---	---

# Lauf­längen­kodierung

internal

a	a	a	F	F	g	G	G	G	a	b	b	b	b	b
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↑  
start

↑  
current

repeat = 2

compressed=

3	a	2	F	g
---	---	---	---	---

# Laufängenkodierung

internal

a	a	a	F	F	g	G	G	G	a	b	b	b	b	b
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↑  
start

↑  
current

repeat = 0

compressed=

3	a	2	F	g
---	---	---	---	---

# Lauf­län­gen­ko­die­rung

internal

a	a	a	F	F	g	G	G	G	a	b	b	b	b	b
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↑  
start  
↑  
current

repeat = 0

compressed=

3	a	2	F	g	3	G	a
---	---	---	---	---	---	---	---

# Lauf­län­gen­ko­die­rung

internal

a	a	a	F	F	g	G	G	G	a	b	b	b	b	b
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↑  
start

↑  
current

repeat = 1

compressed=

3	a	2	F	g	3	G	a
---	---	---	---	---	---	---	---

# Lauf­län­gen­ko­die­rung

internal

a	a	a	F	F	g	G	G	G	a	b	b	b	b	b
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↑  
start

↑  
current

repeat = 2

compressed=

3	a	2	F	g	3	G	a
---	---	---	---	---	---	---	---

# Lauf­län­gen­ko­die­rung

internal

a	a	a	F	F	g	G	G	G	a	b	b	b	b	b
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↑  
start

↑  
current

repeat = 3

compressed=

3	a	2	F	g	3	G	a
---	---	---	---	---	---	---	---

# Laufängerkodierung

internal

a	a	a	F	F	g	G	G	G	a	b	b	b	b	b
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

↑  
start

↑  
current

repeat = 4

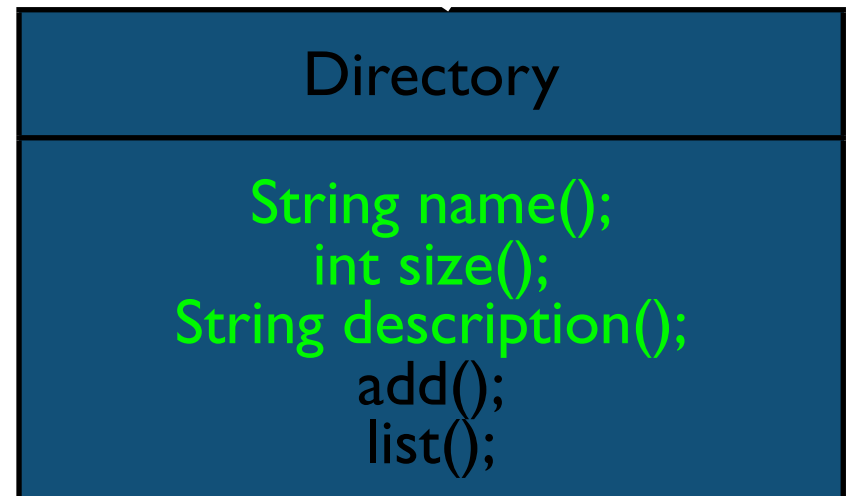
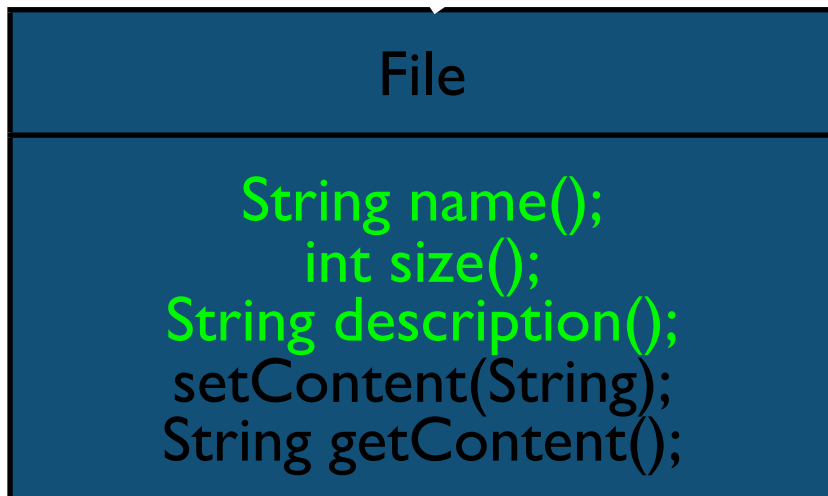
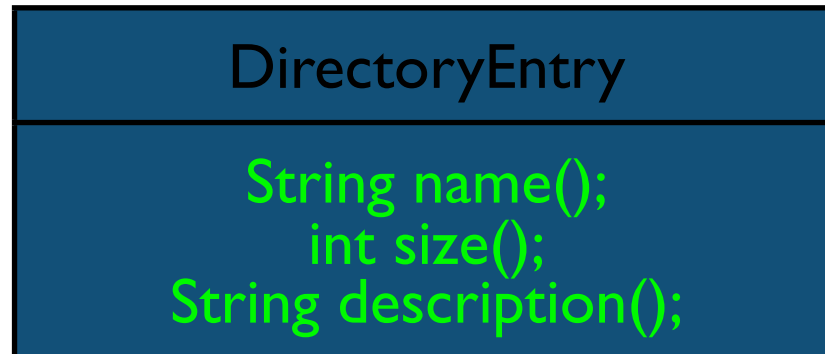
compressed=

3	a	2	F	g	3	G	a	5	b
---	---	---	---	---	---	---	---	---	---

# Beispiel Dateisystem

- Interface: `DirectoryEntry`
- Klassen: `File`, `Directory`

# Beispiel: Dateisystem



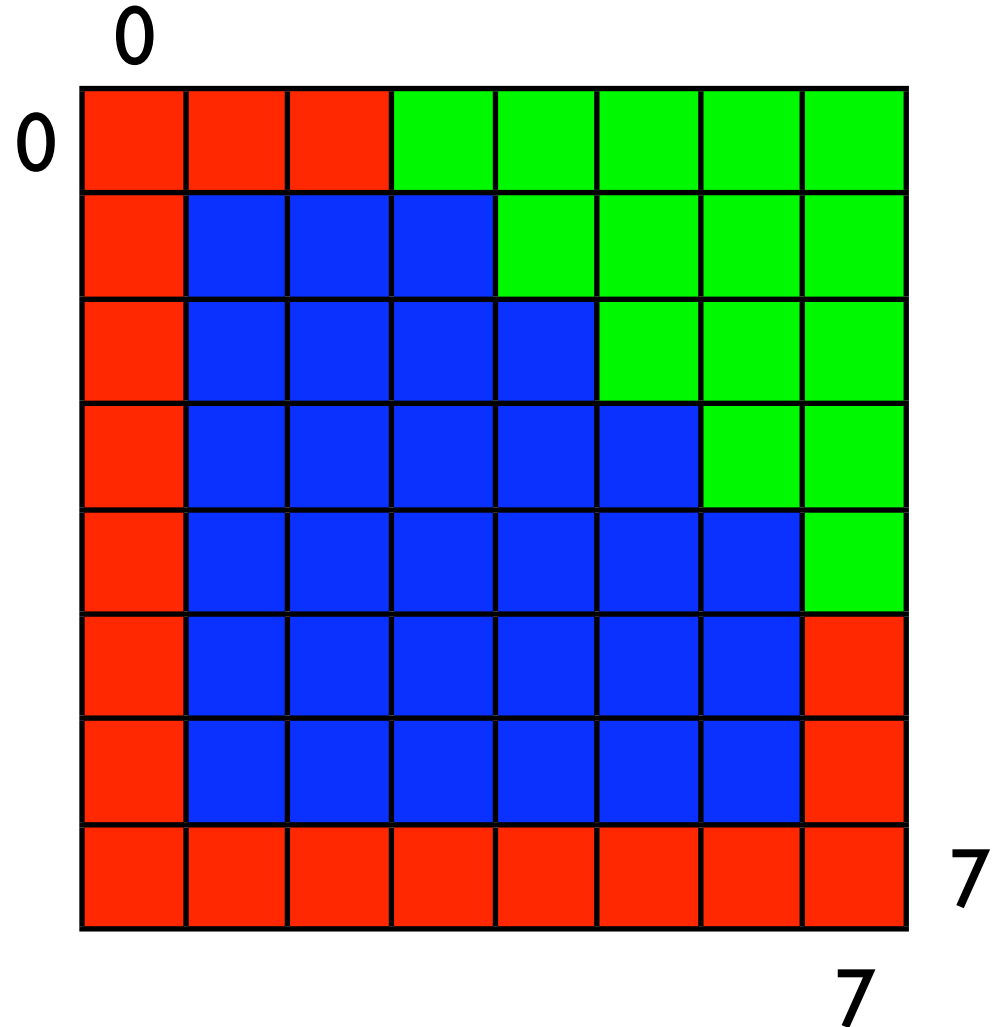
# Beispiel: „Region Growing“

- Implementieren eines Region Growing Algorithmus
- Inhalte:
  - Klassen
  - Umgang mit Arrays
  - Rekursion

```
=====.;+I.:=====
=====;.,.###.;=====
=====,;###.:=====
=====;##W##Y+.;=====
=====.:t##MM##X..=====
=====.:###MMM##MM.;=====
=====;.,;##WMM##.;;=====
=====;..###WWWWM##M,.;=====
=====,.;##WMWMWWW##.;;=====
=====.:XW#WWWWWBWWW#WW.;=====
=====.:B##WW#WWWWWMMW##.;;=====
=====.:Y##WWWMWMWWW#M##.;;=====
=====.:###WWWWMWWWMMW##.;;=====
=====.:.,.###WWWMMWWWMMWBMW#i.======
=====.:t##WWW#####WW#WMM#V.:=====
=====.:##WW#####MMB#Bt.======
=====.:IX#MW##Y=.iiiiitVB##W###.;=====
=====.:W##M##W....;YIii+t#####.;=====
=====.:####MV,...BtiYYI=YY#..t#=:=====
=====.:Y##Y#BRIXRVIYVIIIIiYI#...#..;=====
=====.:#i..MXB+IIIXXXVRXX+XR#...#=:.,=====
=====.:.,.###X##,+IRMRXXXXY,###...t#Y..;=====
=====.:.,;R#...M##VIX+IYIIYR###...###.;=====
=====.:W##.....,i#####RRR#####.:.,.XM##X;.,=====
=====.:.,.+###.....,IB#####MVi.....,=I##XV#...=
=====.:.,.=###MV.....;it;=.....,+W;.V##...
=====.:X##..##I.....,.,.#+....Y#:.
=====.:##t...i#t.....,.,.Y#.=...#Y
=====.:#t...:i#i.....:.....#+...#i
```

# „Region Growing“

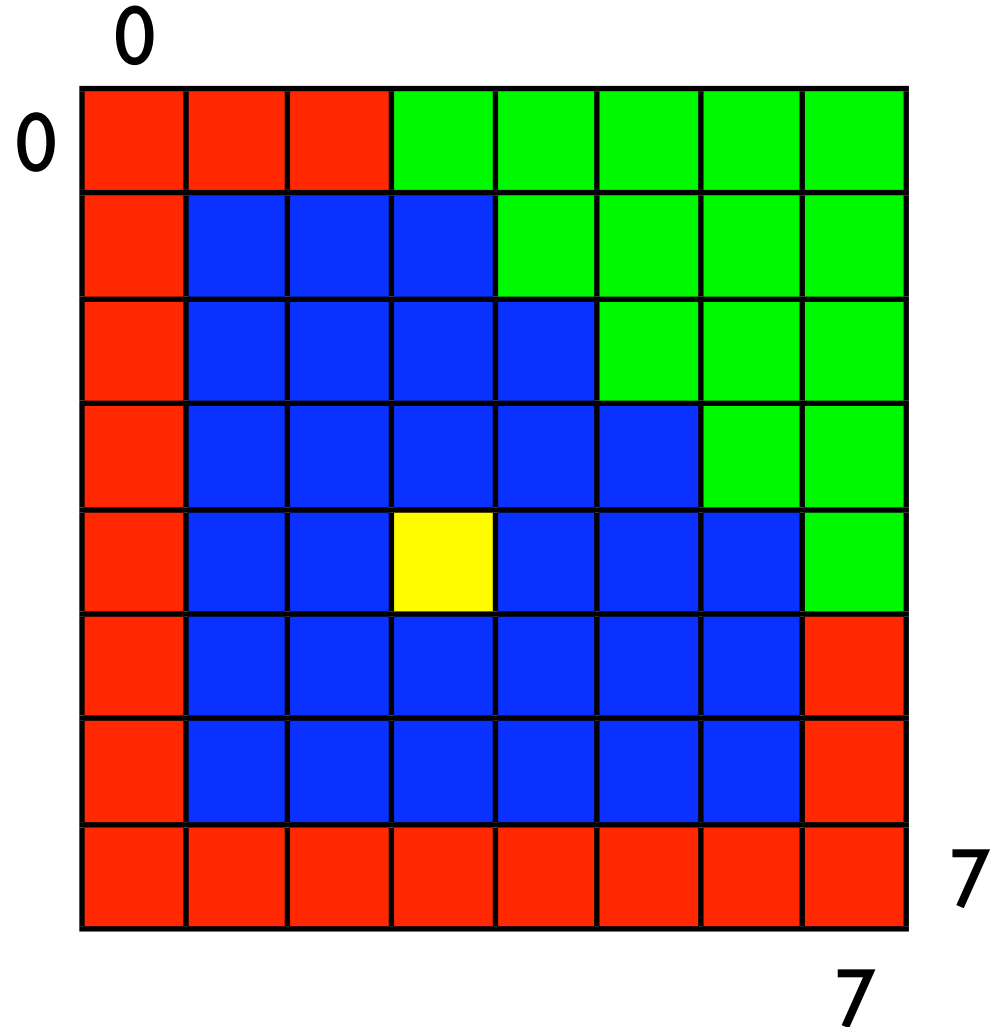
Befüllen von  
Regionen mit  
gleichem Farbwert  
mit neuer Farbe



# „Region Growing“

`fill(3,4,yellow)`

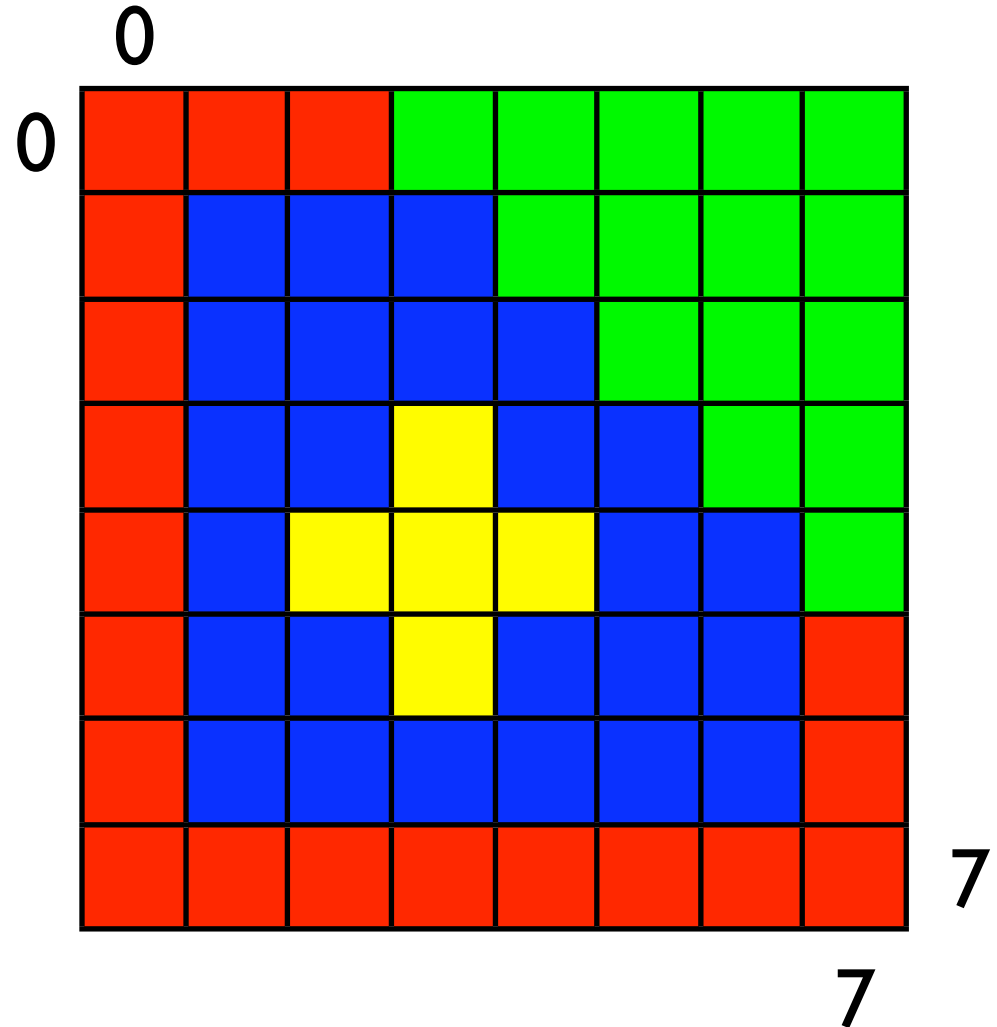
Ausgehend vom Punkt  
(3,4) werden  
benachbarte Pixel mit  
gleichem Farbwert gelb  
gefärbt



# „Region Growing“

`fill(3,4,yellow)`

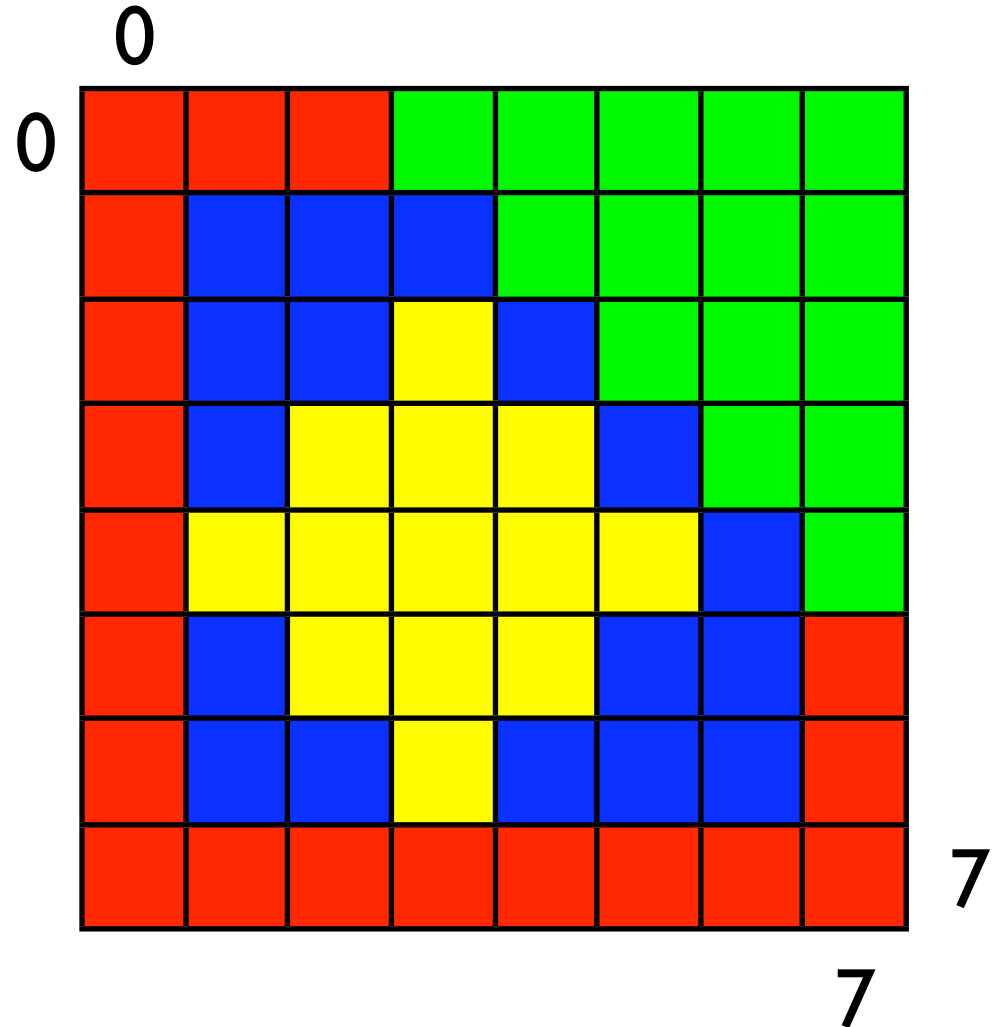
Ausgehend vom Punkt  
(3,4) werden  
benachbarte Pixel mit  
gleichem Farbwert gelb  
gefärbt



# „Region Growing“

`fill(3,4,yellow)`

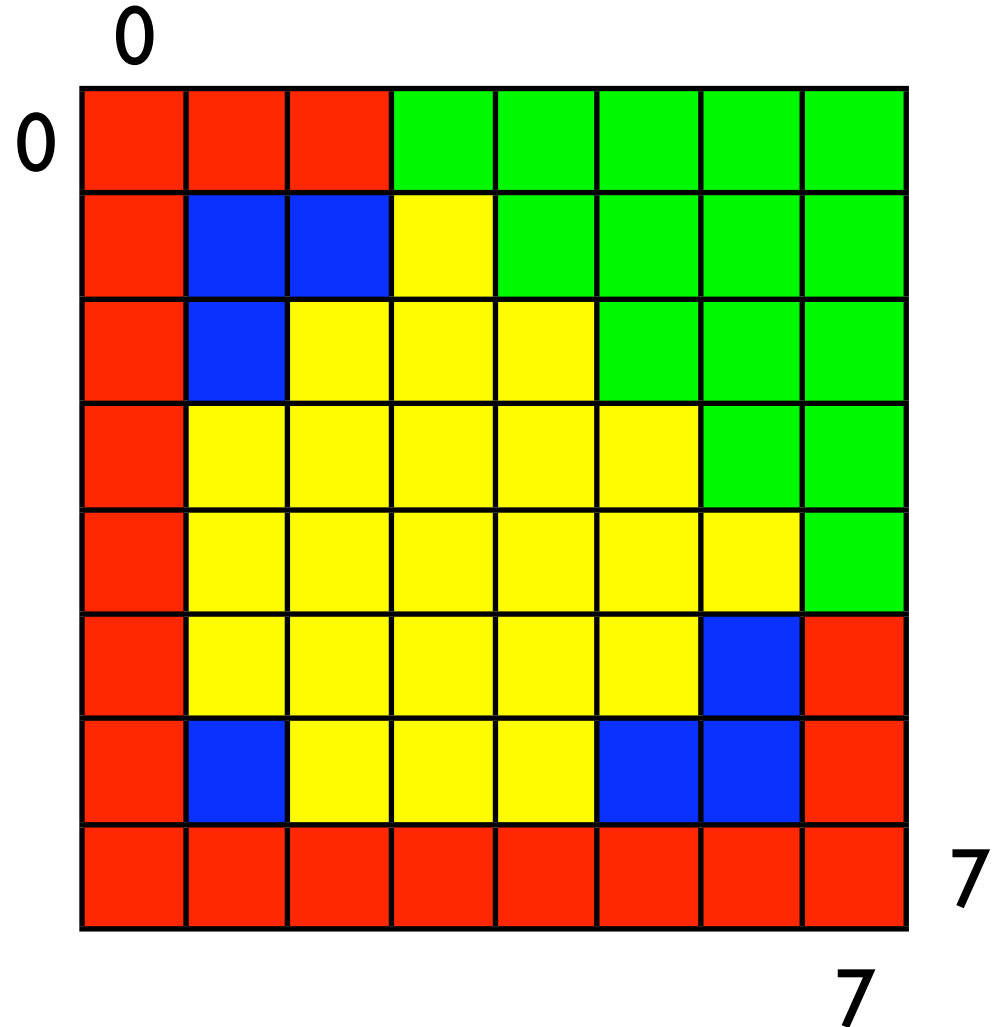
Ausgehend vom Punkt  
(3,4) werden  
benachbarte Pixel mit  
gleichem Farbwert gelb  
gefärbt



# „Region Growing“

`fill(3,4,yellow)`

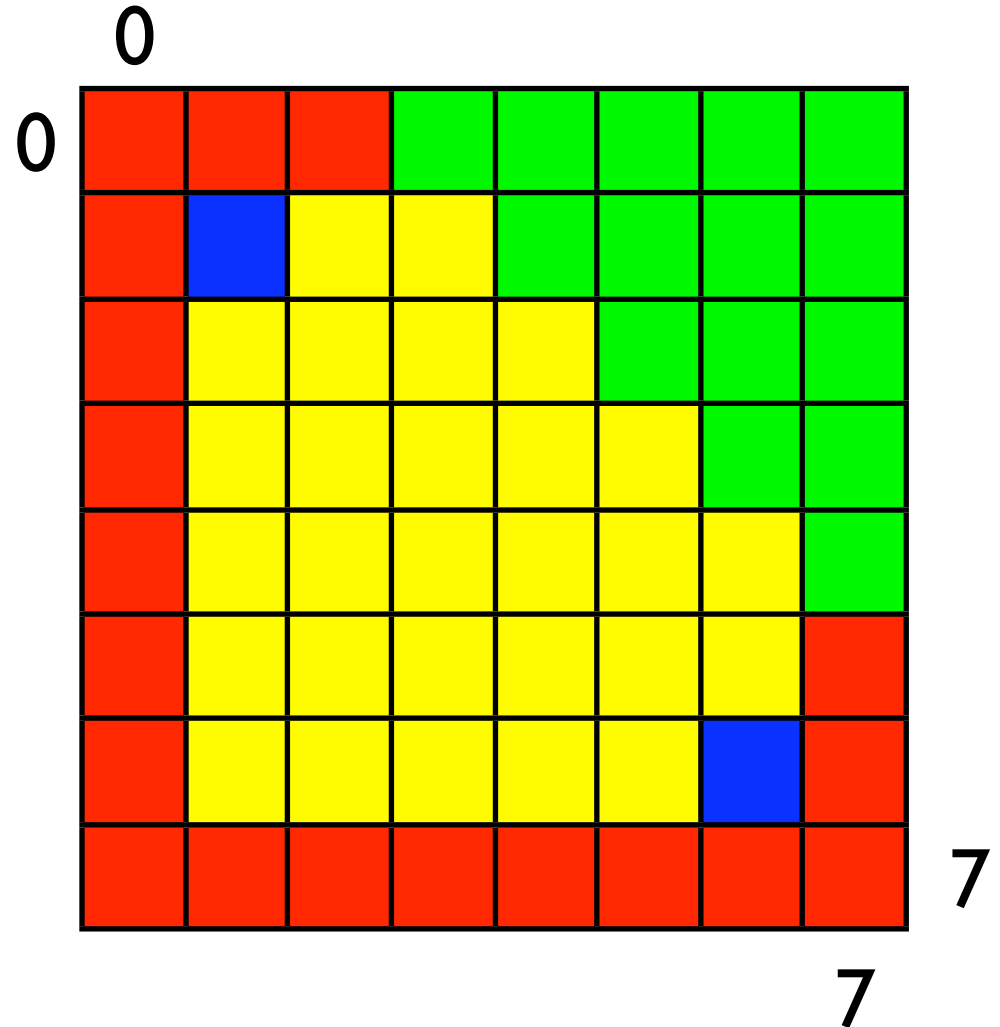
Ausgehend vom Punkt  
(3,4) werden  
benachbarte Pixel mit  
gleichem Farbwert gelb  
gefärbt



# „Region Growing“

`fill(3,4,yellow)`

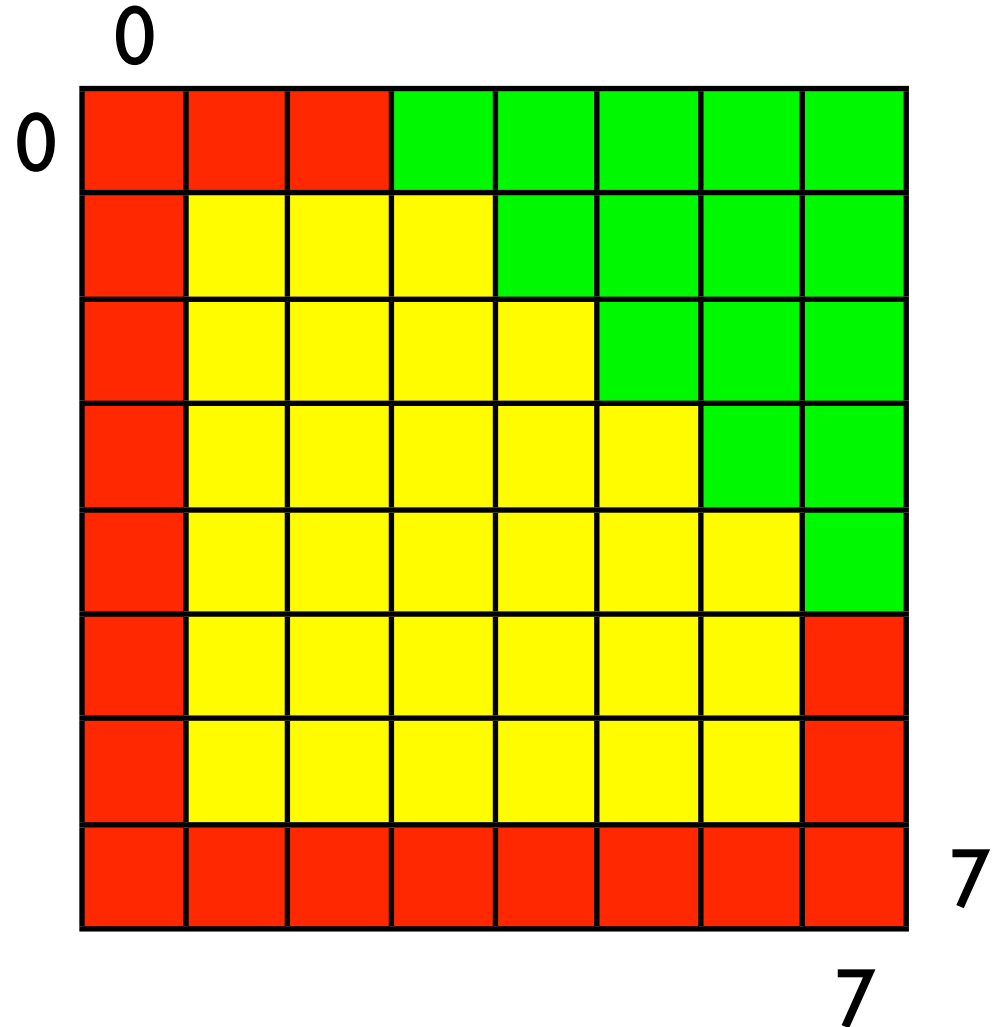
Ausgehend vom Punkt  
(3,4) werden  
benachbarte Pixel mit  
gleichem Farbwert gelb  
gefärbt



# „Region Growing“

`fill(3,4,yellow)`

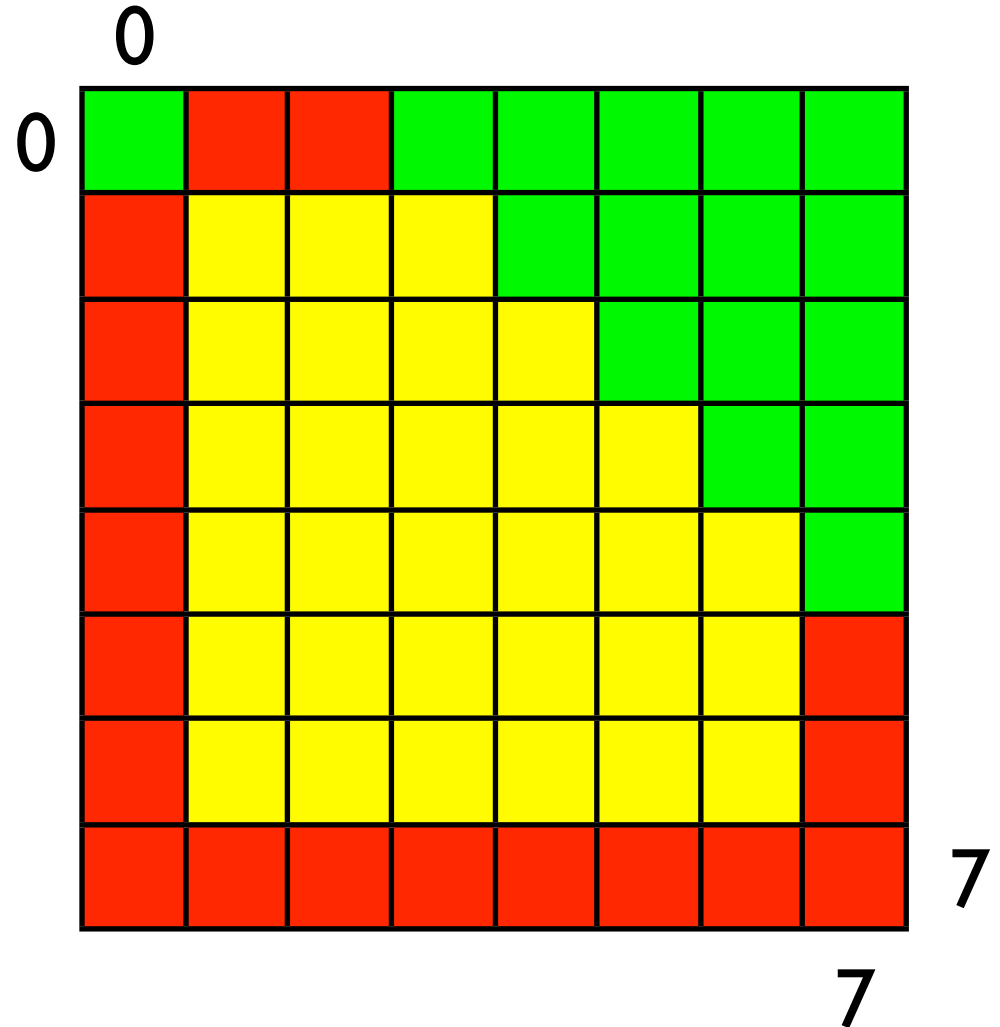
Ausgehend vom Punkt  
(3,4) werden  
benachbarte Pixel mit  
gleichem Farbwert gelb  
gefärbt



# „Region Growing“

`fill(3,4,yellow)`

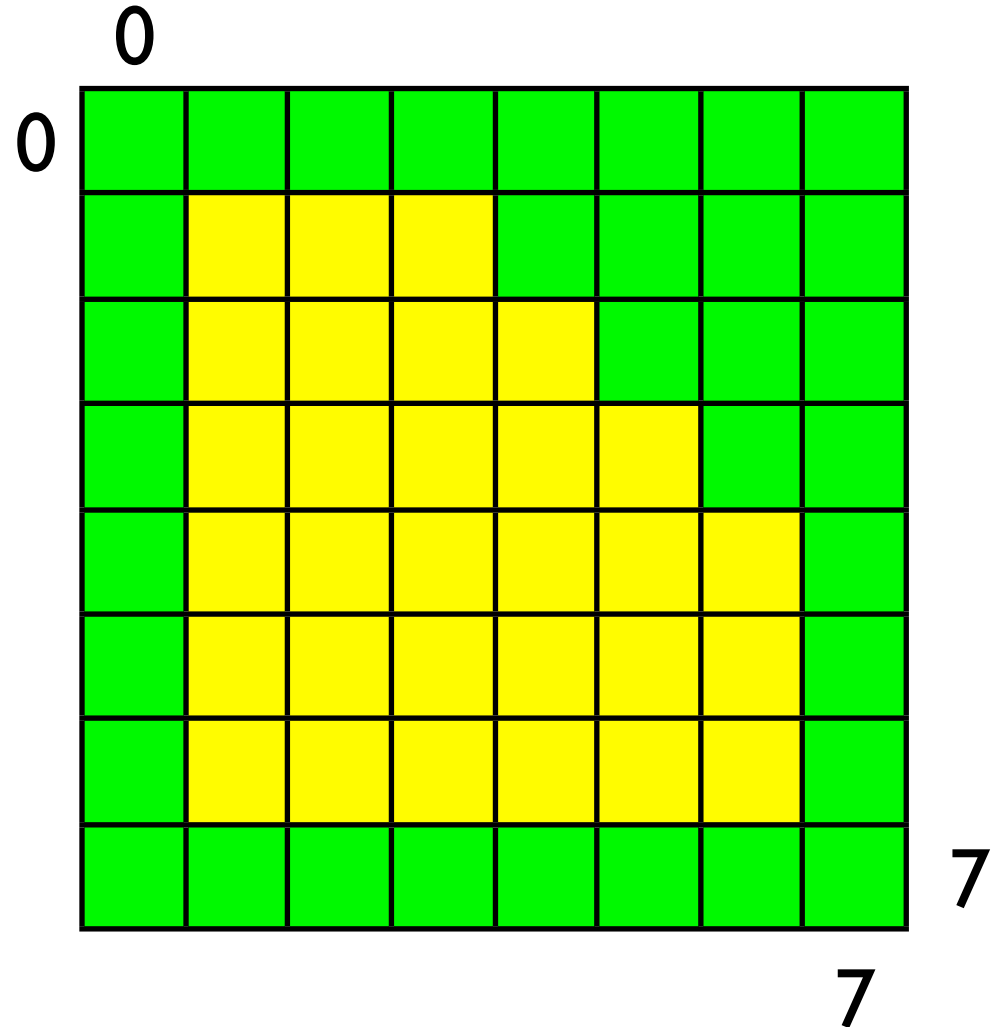
`fill(0,0,green)`



# „Region Growing“

`fill(4,5,yellow)`

`fill(0,0,green)`

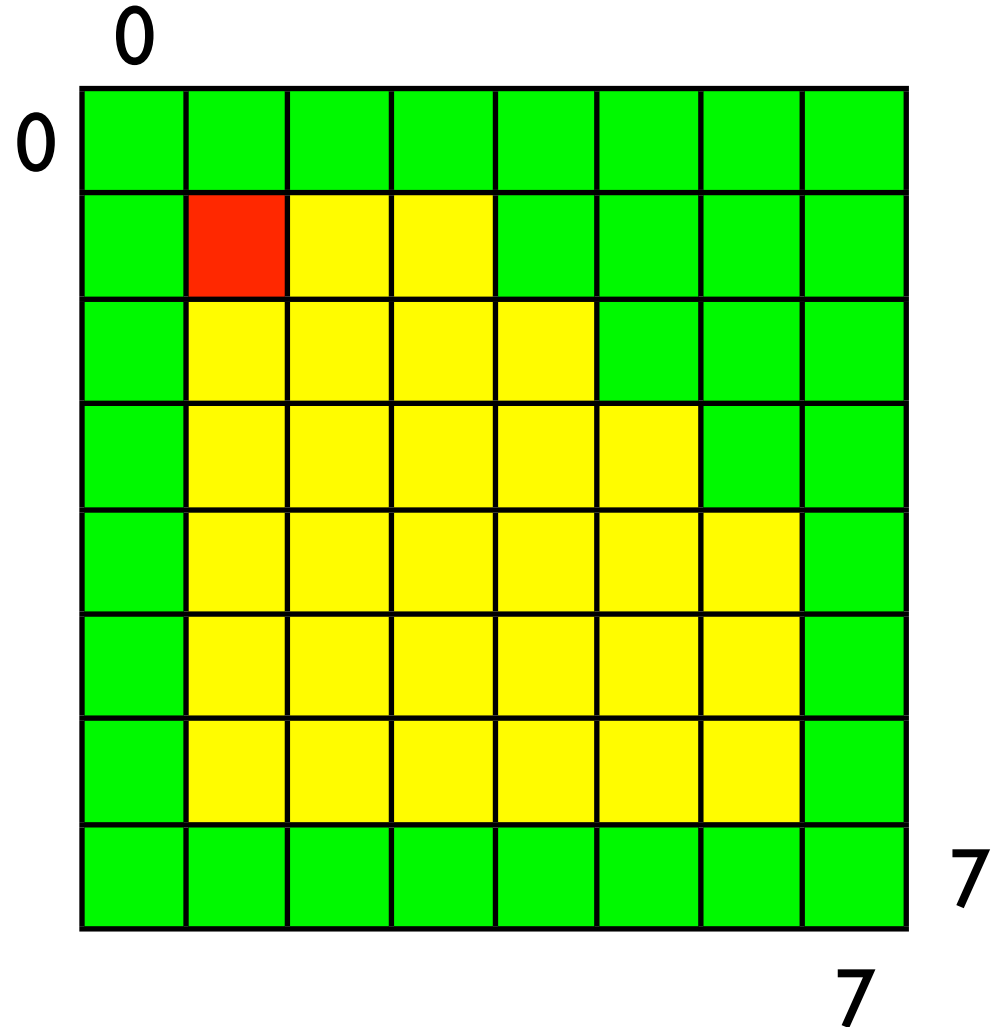


# „Region Growing“

`fill(3,4,yellow)`

`fill(0,0,green)`

`fill(1,1,red)`

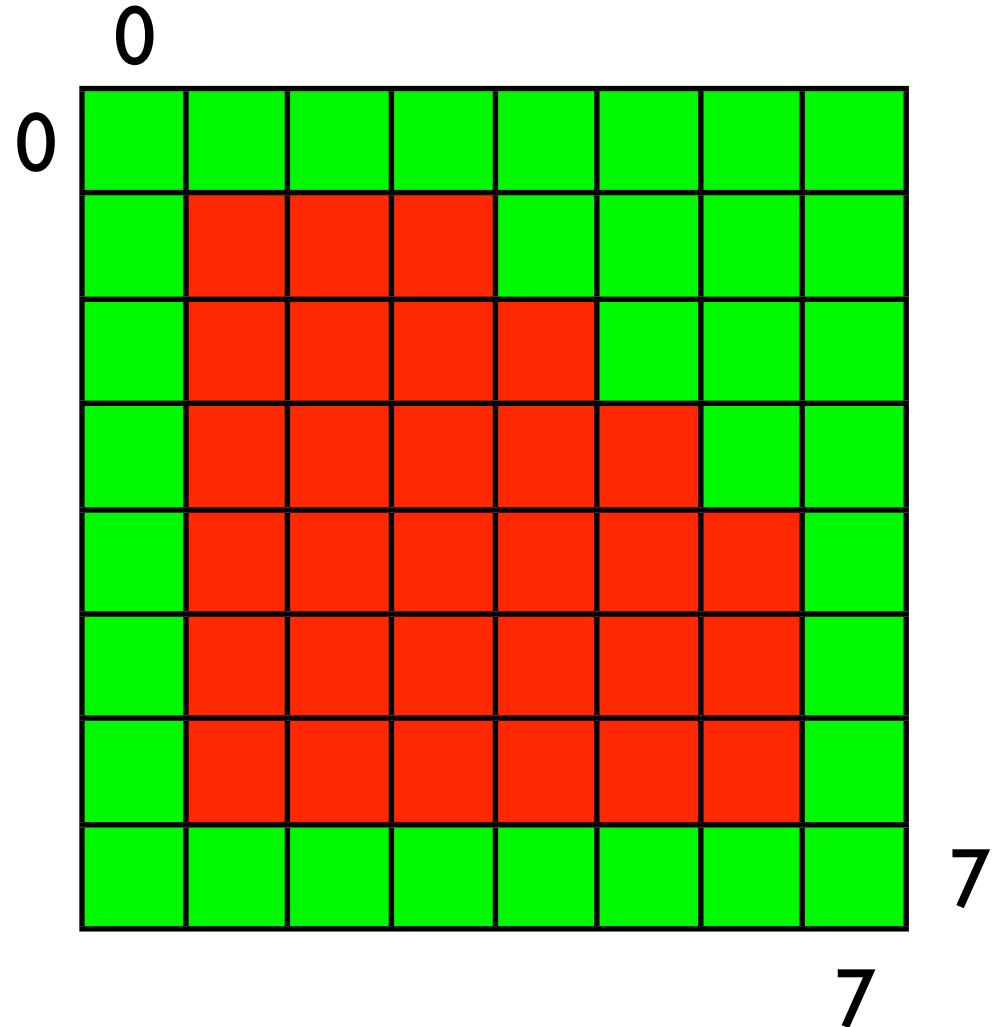


# „Region Growing“

fill(3,4,yellow)

fill(0,0,green)

fill(1,1,red)



# Rekursiver Aufruf

```
internal[i][j] = put;
```

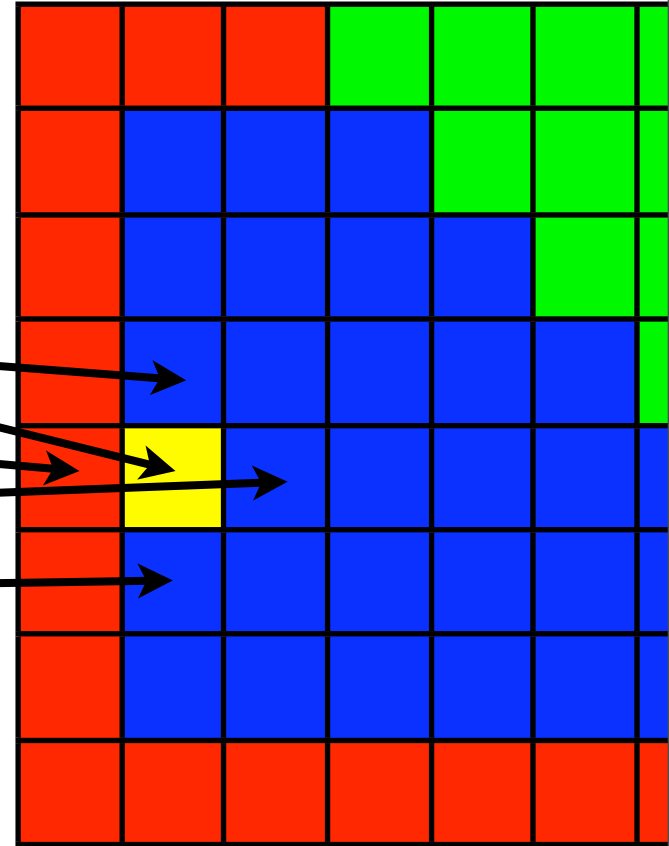
```
return fill(i-1,j,put,replace)
```

```
+ fill(i,j-1,put,replace)
```

```
+ fill(i,j+1,put,replace)
```

```
+ fill(i+1,j,put,replace)
```

```
+ 1;
```



# Rekursiver Aufruf

```
internal[i][j] = put;
```

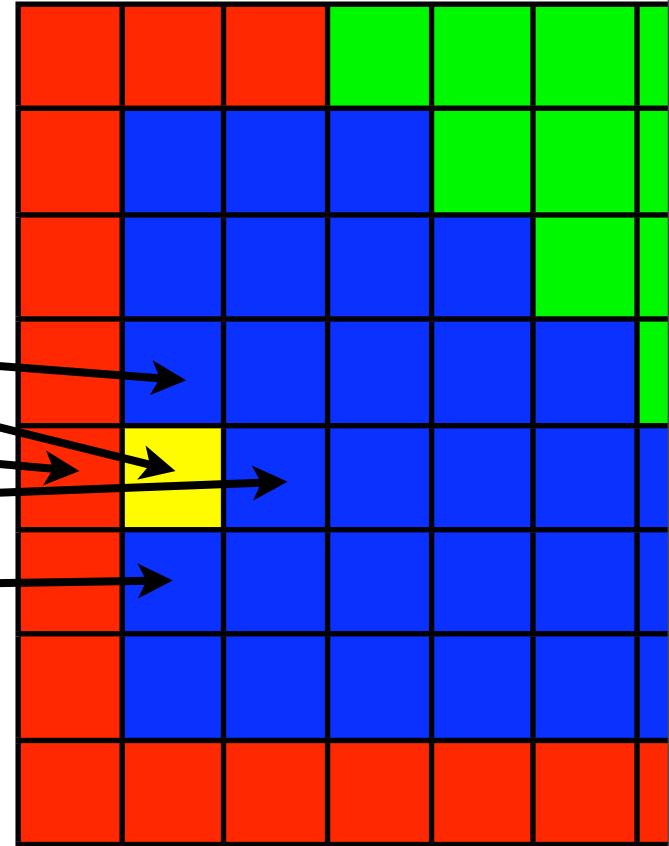
```
return fill(i-1,j,put,replace)
```

```
+ fill(i,j-1,put,replace)
```

```
+ fill(i,j+1,put,replace)
```

```
+ fill(i+1,j,put,replace)
```

```
+ 1;
```



# Rekursiver Aufruf

```
if (internal[i][j] == replace )  
{  
    internal[i][j] = put;  
  
    return fill(i-1,j,put,replace)  
    + fill(i,j-1,put,replace)  
    + fill(i,j+1,put,replace)  
    + fill(i+1,j,put,replace)  
    + 1;  
} else return 0;
```

