

Algorithmen und Datenstrukturen (ESE)
Entwurf, Analyse und Umsetzung von
Algorithmen (IEMS)
WS 2014 / 2015

Vorlesung 12, Donnerstag, 22. Januar 2015
(Graphen, Breiten/Tiefensuche, Zusammenhangskomponenten)

Junior-Prof. Dr. Olaf Ronneberger
Image Analysis Lab
Institut für Informatik
Universität Freiburg

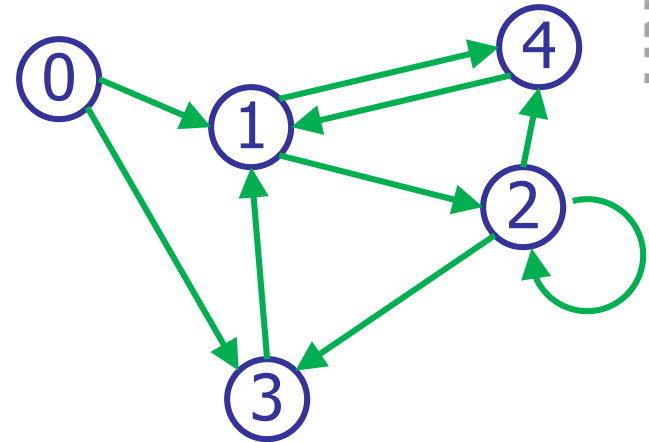
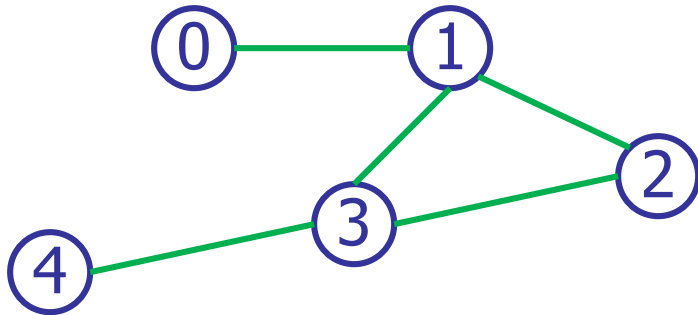
■ Graphen

- Neben Feldern, Listen und Bäumen die häufigste Datenstruktur (Bäume sind eine spezielle Art von Graph)
- Darstellung im Rechner
- **Breitensuche** (Breadth First Search = **BFS**)
- **Tiefensuche** (Depth First Search = **DFS**)
- **Zusammenhangskomponenten** eines Graphen
- **Übungsblatt 12:** Berechnung der größten Zusammenhangskomponente in einem Straßengraphen mittels **BFS** oder **DFS**

Rückmeldungen zum Ü11 (Beweis $O(n)$ für (4,9) Bäume)

- Zeitaufwand bei fast allen deutlich unter 4 Std.
- „gut machbar“, „nicht schwierig“, „Quite useful exercise since it really requires an insight in how things are done“
- „schwierig die Vorlesungsfolien zu verstehen und auch die Videoaufzeichnungen halfen mir nicht viel“ → direkt in der Vorlesung fragen
- „Leider wieder nur ein Beweis und kein Code.“ → heute wieder programmieren

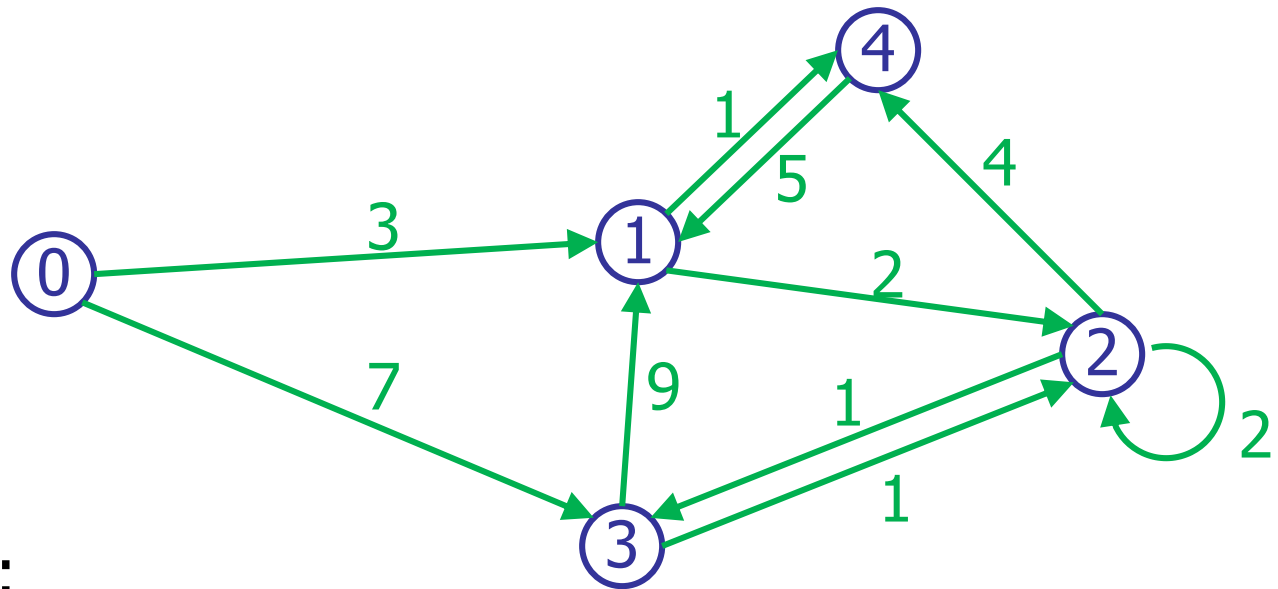
Graphen: Definition



■ Definition:

- Ein Graph $G = (V, E)$ besteht aus einer Menge V von Knoten ...
 - Englisch: **vertices** (daher V) oder **nodes**
- ... und einer Menge E von Kanten
 - Englisch: **edges** (daher E) oder **arcs**
- Eine Kante e verbindet jeweils zwei Knoten u und v
 - ungerichtete Kante: $e = \{u, v\}$ (Menge)
 - gerichtete Kante: $e = (u, v)$ (Tupel)
- Es kann auch „self-loops“ geben: (u, u)

Gewichteter Graph



■ Definition:

- **Gewichteter** Graph: Eine reelle Zahl pro Kante, das sogenannte **Gewicht** der Kante, je nach Anwendung auch **Länge** oder **Kosten** der Kante genannt
- Beispiel: Straßennetz.
 - Kreuzungen: **Knoten**
 - Straßen: **Kanten**
 - Fahrzeit: **Kosten der Kante**

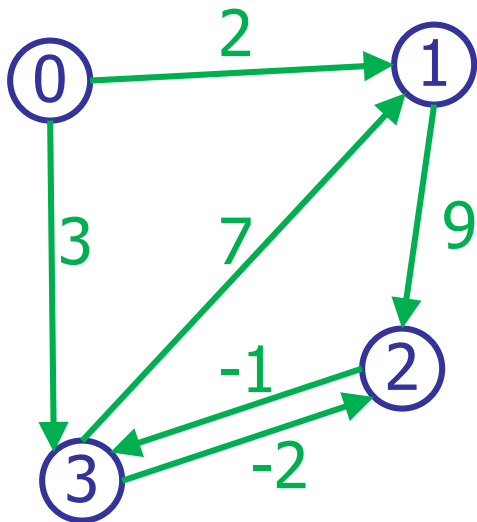
Repräsentation im Rechner

■ Wie repräsentiert man Graphen im Rechner

- Da gibt es zwei klassischen Arten

Adjazenzmatrix ... Platzverbrauch $\Theta(|V|^2)$

Adjazenzlisten bzw. **–felder** ... Platzverbrauch $\Theta(|V| + |E|)$



**Gerichteter Graph
mit Kantengewichten**

$$|V| = 4, |E| = 6$$

		Zielknoten			
		0	1	2	3
Startknoten	0		2		3
	1			9	
	2				-1
	3		7	-2	

Adjazenzmatrix

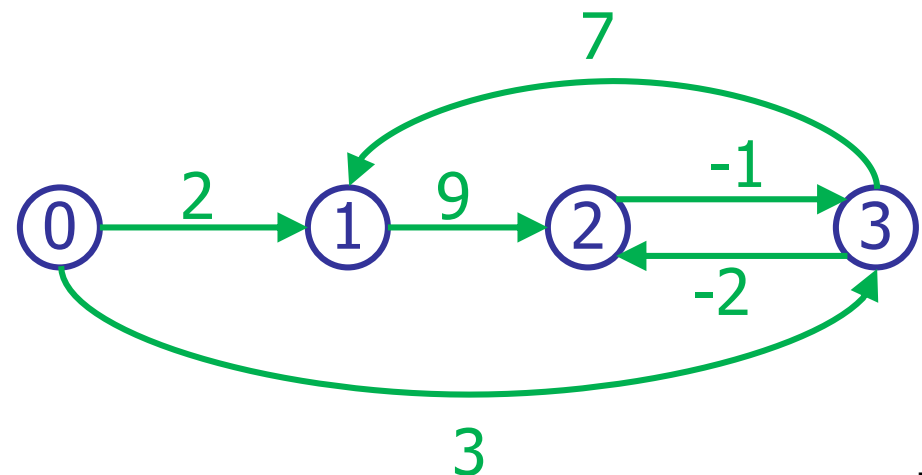
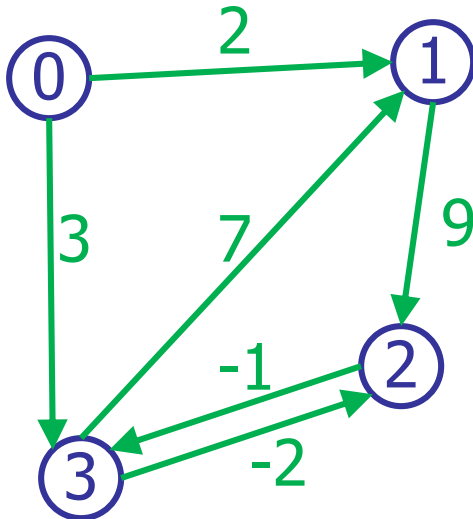
		Zielknoten	Kosten
Startknoten	0	1, 2	3, 3
	1	2, 9	
	2	3, -1	
	3	1, 7	2, -2

Adjazenzlisten

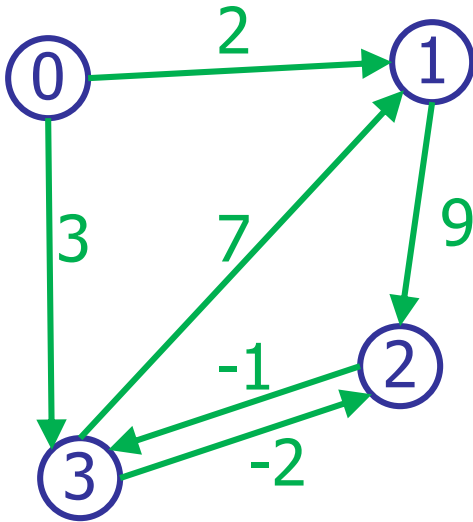
Graphen: Anordnung

- Der Graph ist durch die Adjazenzmatrix, bzw. durch die Adjazenzlisten eindeutig definiert.
- Die Anordnung bei der Zeichnung des Graphen spielt keine Rolle:

Startknoten	① 0	1, 2	3, 3
	① 1	2, 9	
	① 2	3, -1	
	① 3	1, 7	2, -2



Graphen Implementierung



	Zielknoten	Kosten
Startknoten ① 0	1, 2	3, 3
① 1	2, 9	
② 2	3, -1	
③ 3	1, 7	2, -2

Adjazenzlisten

```
class Arc {
public:
    int headNodeId;
    int cost;
};
```

- Implementierung z.B. als Feld von Feldern
 - **C++:** `std::vector< std::vector<Arc> > adjacency_lists;`
 - **Java:** `ArrayList<ArrayList<Arc>> adjacencyLists;`
- Oder als Feld von verketteten Listen
 - **C++:** `std::vector< std::list<Arc> > adjacency_lists;`
 - **Java:** `ArrayList<LinkedList<Arc>> adjacencyLists;`

Implementierung: GraphTest.cpp

```
#include <gtest/gtest.h>
#include "../Graph.h"

TEST(GraphTest, constructor) {
    Graph graph;
    ASSERT_EQ("{0, 0}", graph.toString());
}

TEST(GraphTest, addNodesAndArcs) {
    Graph graph;
    // Add three nodes: 0, 1, 2.
    graph.addNode();
    graph.addNode();
    graph.addNode();
    // Add four arcs.
    graph.addArc(0, 1);
    graph.addArc(1, 2);
    graph.addArc(0, 2);
    graph.addArc(2, 0);
    // Check that the correct graph was constructed.
    ASSERT_EQ("{3, 4, (0,1), (0,2), (1,2), (2,0)}", graph.toString());
}
```

Implementierung: Graph.h

```
#include <string>

class Graph {
public:
    // Construct an empty graph.
    Graph();

    // Add a node to this graph.
    void addNode();

    // Add an arc = a directed edge to this graph.
    void addArc(int u, int v);

    // Show the graph in human-readable form
    // (useful for debugging and testing).
    std::string toString();
};
```

Implementierung: Graph.cpp

```
#include "../Graph.h"
```

```
Graph::Graph() {  
}
```

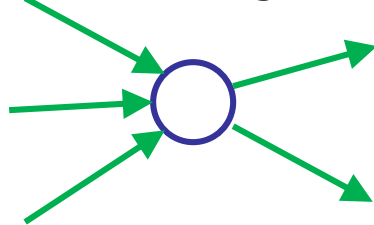
```
void Graph::addNode() {  
}
```

```
void Graph::addArc(int u, int v) {  
}
```

```
std::string Graph::toString() {  
    return "";  
}
```

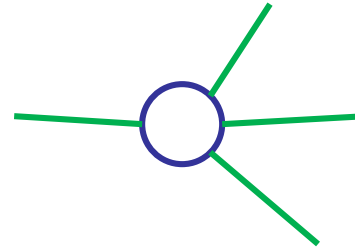
Graphen: Eingangs- / Ausgangsgrad

Eingangs-
grad 3



Ausgangs-
grad 2

Grad 4



■ Grade in einem Graphen $G = (V, E)$

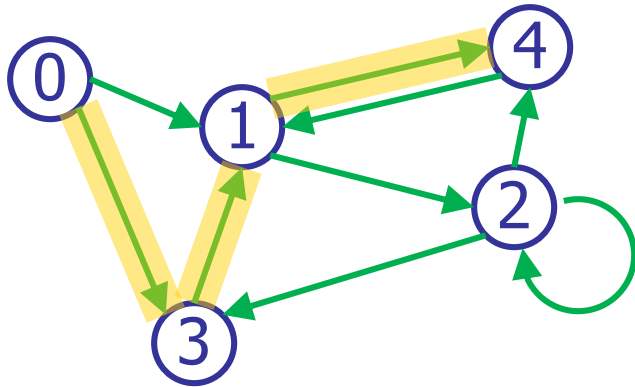
– Falls gerichtet

- **Eingangsgrad** von einem Knoten u
= Anzahl eingehender Kanten = $|\{(v,u) : (v,u) \in E\}|$
- **Ausgangsgrad** von einem Knoten u
= Anzahl ausgehender Kanten = $|\{(u,v) : (u,v) \in E\}|$

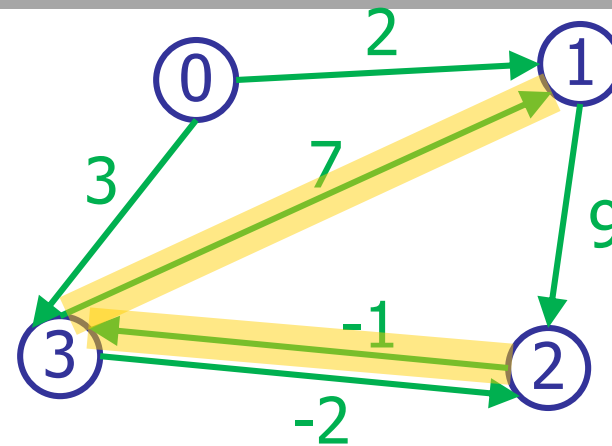
– Falls ungerichtet

- **Grad** von einem Knoten u
= Anzahl adjazenter Kanten = $|\{\{u,v\} : \{u,v\} \in E\}|$

Pfade im Graphen



Pfad 0,3,1,4 mit Länge 3

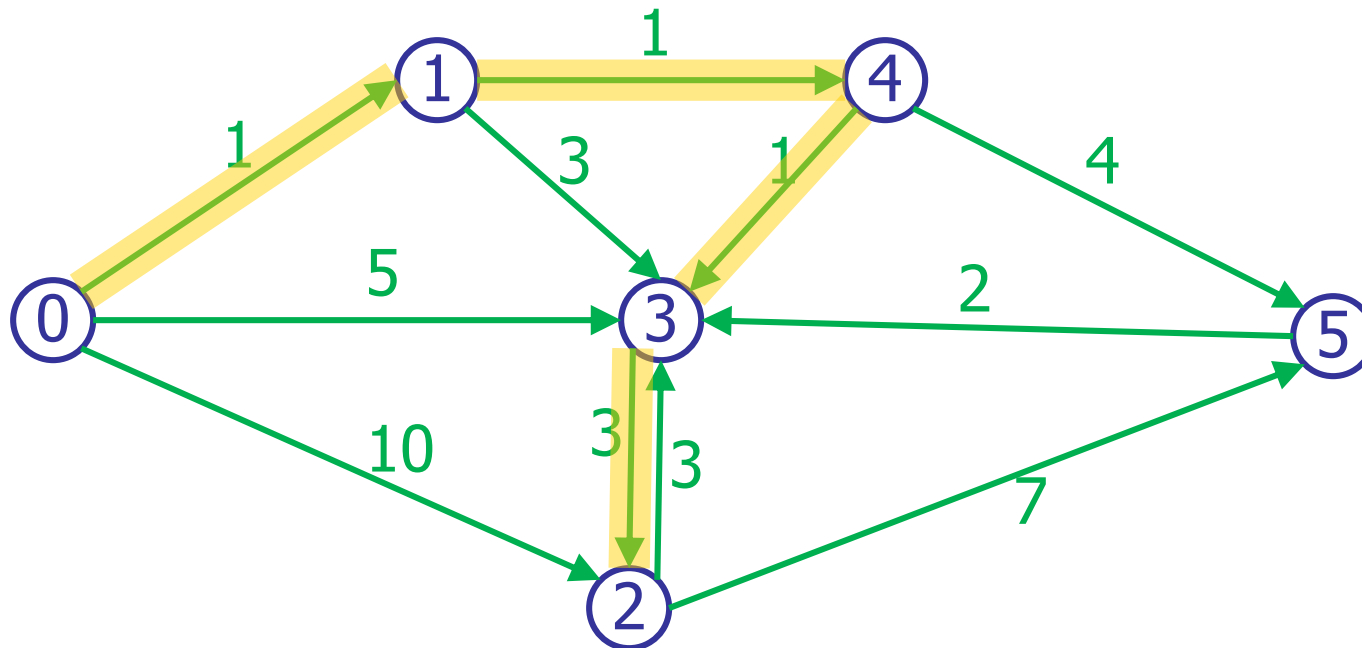


Pfad 2,3,1 mit Kosten 6

■ Pfade in einem Graphen $G = (V, E)$

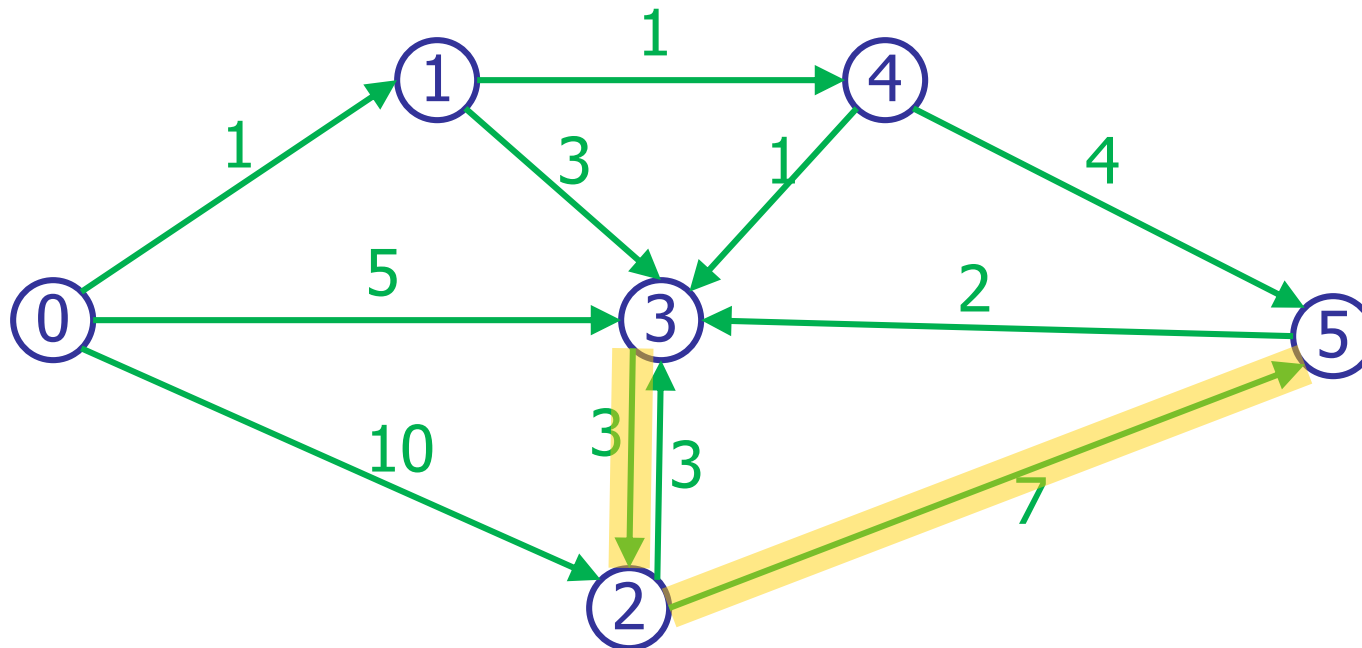
- Ein Pfad in G ist eine Folge $u_1, u_2, u_3, \dots, u_l \in V$ mit
 - $(u_1, u_2), (u_2, u_3), \dots, (u_{l-1}, u_l) \in E$ [gerichteter Graph]
 - $\{u_1, u_2\}, \{u_2, u_3\}, \dots, \{u_{l-1}, u_l\} \in E$ [ungerichteter Graph]
- Die **Länge des Pfades** (auch: Kosten des Pfades)
 - ohne Kantengewichte: Anzahl der Kanten
 - mit Kantengewichten: Summe der Gewichte auf dem Pfad

Kürzester Pfad



■ Pfade in einem Graphen $G = (V, E)$

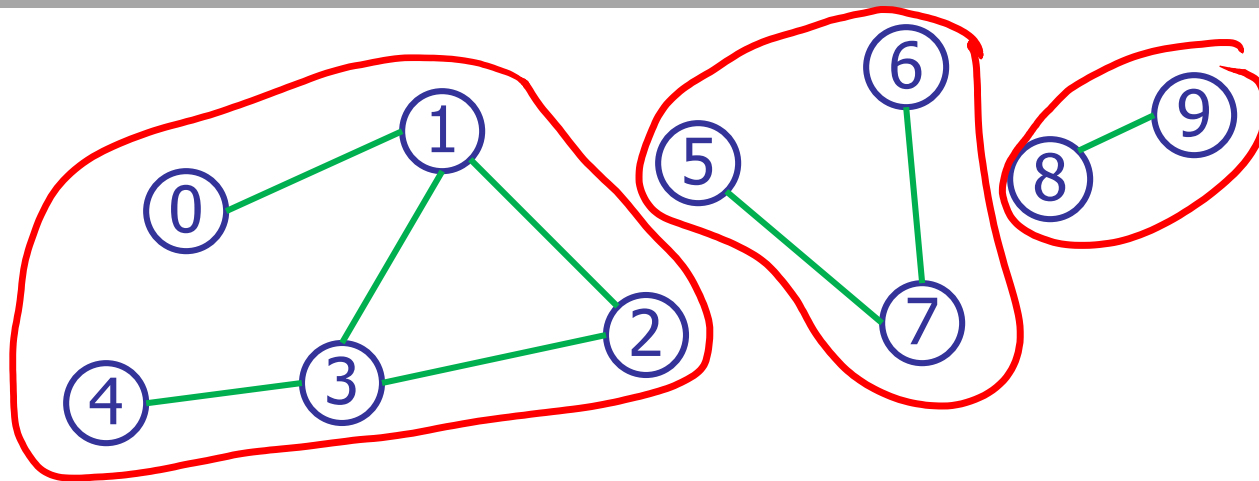
- Der **kürzeste Pfad** (engl. *shortest path*) zwischen zwei Knoten u und v ist der Pfad u, \dots, v mit der kürzesten Länge (bzw. geringsten Kosten)
- Beispiel: Der kürzeste Pfad von 0 nach 2?
 - Pfad 0,1,4,3,2 mit Kosten 6



■ Pfade in einem Graphen $G = (V, E)$

- Der **Durchmesser** eines Graphen ist der längste kürzeste Pfad = $\max_{u,v} \{\text{Länge von } P : P \text{ ist ein kürzester Pfad zwischen } u \text{ und } v\}$
- Durchmesser des Beispielgraphen?
 - Knoten 3 und 5 sind „am weitesten“ entfernt: Kosten 10

Zusammenhangskomponenten



- Für einen **ungerichteten** Graphen $G = (V, E)$
 - Die Zusammenhangskomponenten bilden eine Partition von V , also $V = V_1 \cup V_2 \cup \dots \cup V_k$
 - Zwei Knoten u und v sind in derselben Zusammenhangskomponente, wenn es einen Pfad zwischen u und v gibt

(Für **gerichtete** Graphen ist die Definition komplizierter, man spricht dann von **starken** Zusammenhangskomponenten, das machen wir in dieser Vorlesung aber nicht)

■ Informale Definition

- Gegeben ein Graph $G = (V, E)$ und ein Startknoten $s \in V$, besuche "systematisch" alle Knoten von V , die von s erreichbar sind
- Breitensuche = in der Reihenfolge der "Entfernung" von s
 - Englisch: **breadth first search = BFS**
- Tiefensuche = erstmal "möglichst weit weg" von s
 - Englisch: **depth first search = DFS**
- Das ist kein "Problem" an sich, taucht aber oft als Teil / Subroutine von anderen Algorithmen auf

Zum Beispiel in der Übungsaufgabe, zur Berechnung der Zusammenhangskomponenten

■ Idee

- Markierung für jeden Knoten, zu Beginn alle unmarkiert
- Beginne mit einem **Startknoten** und markiere ihn (**Level 0**)
- Finde alle Knoten die zum **Startknoten** benachbart und noch nicht markiert sind und markiere sie (**Level 1**)
- Finde alle Knoten, die zu einem **Level-1** Knoten benachbart und noch nicht markiert sind und markiere sie (**Level 2**)
- Usw. bis ein Level keine benachbarten Knoten mehr hat, die noch nicht markiert sind
- Das markiert insbesondere alle Knoten, die in derselben **Zusammenhangskomponente** sind wie der Startknoten

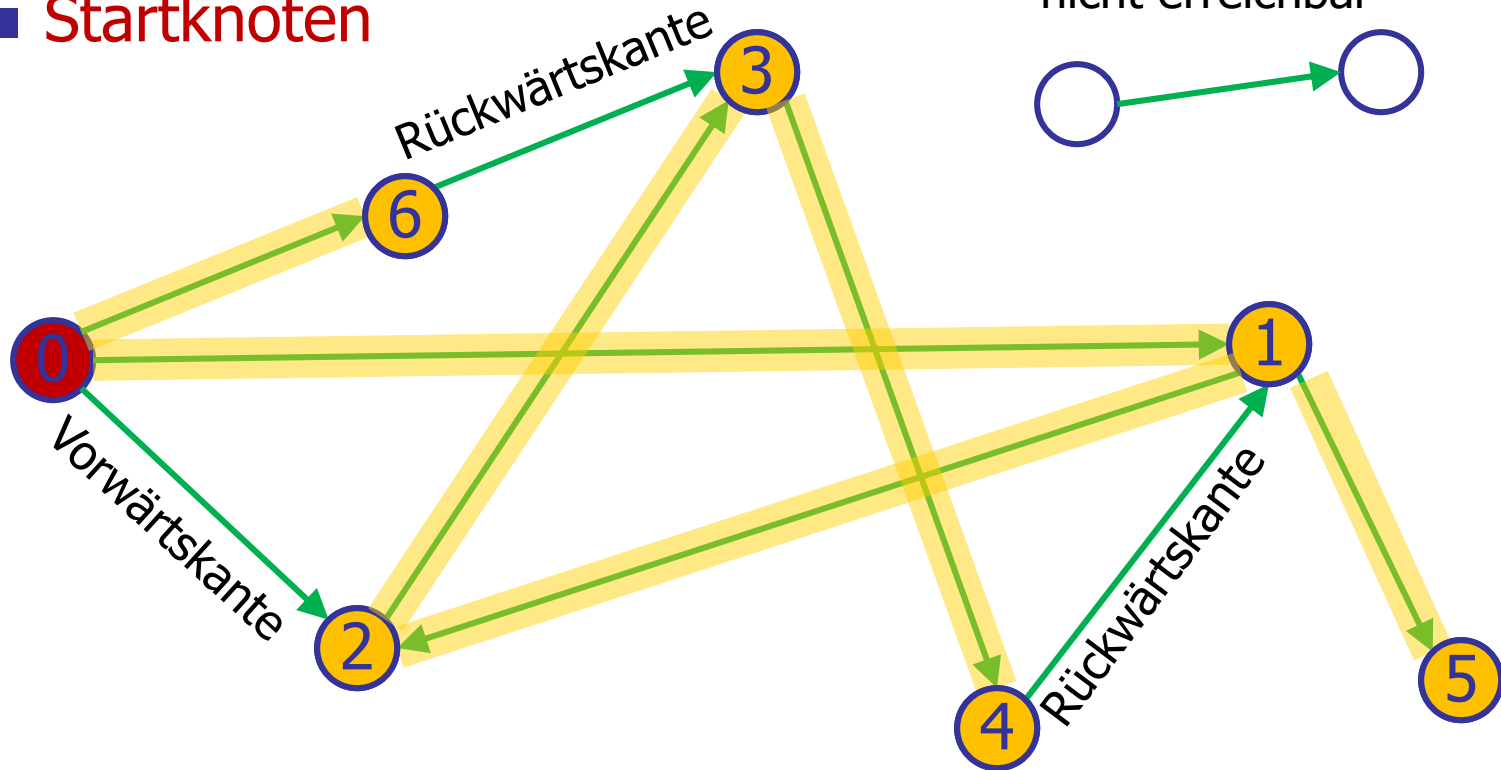
Tiefensuche (DFS) 1/2

■ Idee

- Markierung für jeden Knoten, zu Beginn alle unmarkiert
- Beginne mit einem **Startknoten** und markiere ihn
- Gehe in irgendeiner Reihenfolge die zum Startknoten benachbarten Knoten durch und tue Folgendes:
Falls der Knoten noch nicht markiert ist, markiere ihn und starte **rekursiv** eine Tiefensuche von dort aus
- Das sucht zuerst "in die Tiefe" (vom Startknoten aus)
- Auch **DFS** markiert schließlich alle Knoten, die in derselben Zusammenhangskomponenten liegen wie der Startknoten
- Auf azyklischen Graphen liefert **DFS topologische Sortierung**
Das ist eine Nummerierung der Knoten, so dass jede Kante von einem Knoten mit kleinerer Nummer zu einem mit größerer Nummer geht

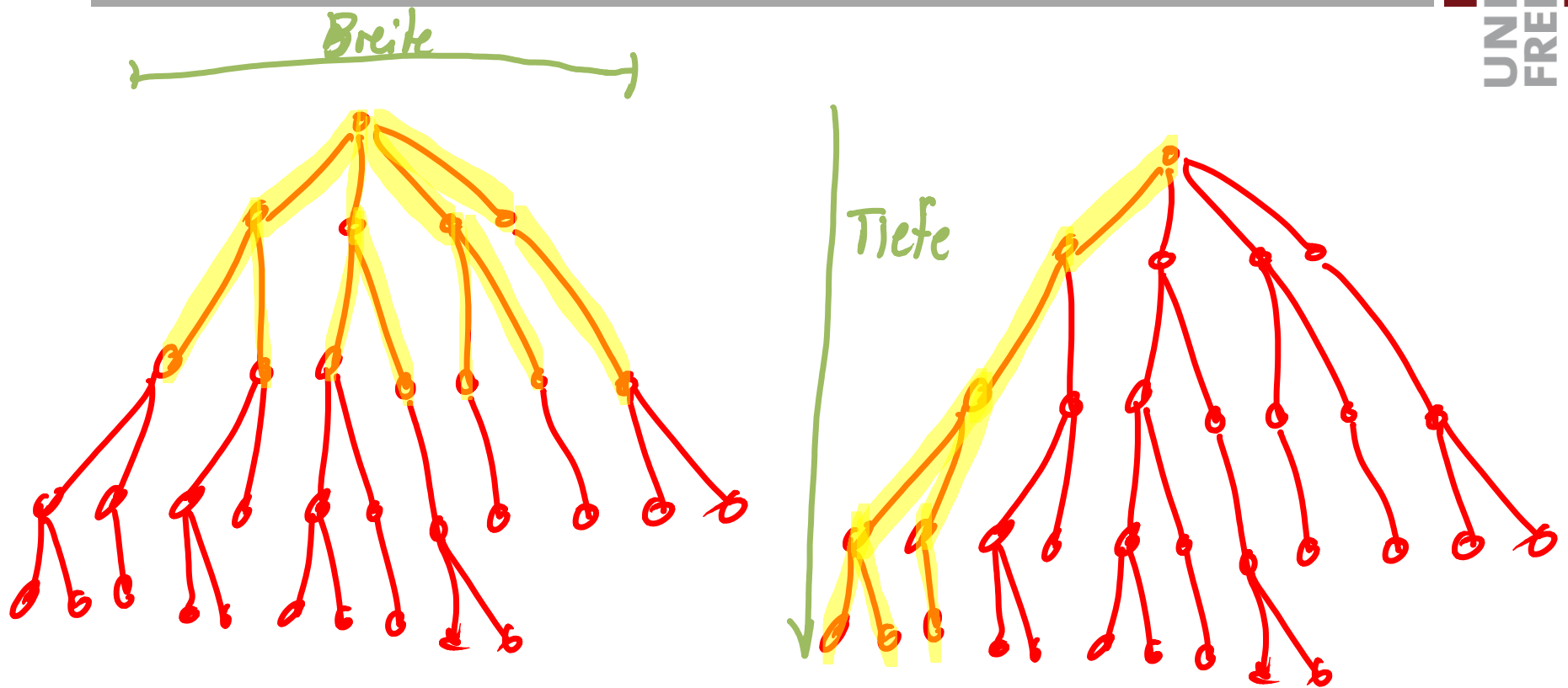
Tiefensuche (DFS) 1/2

■ Startknoten



- Die markierten Kanten bilden wieder einen aufspannenden Baum, aber einen anderen
- Wenn der Graph azyklisch ist (ohne Kanten (6,3) und (4,1)), dann liefert die Nummerierung eine **topologische Sortierung**

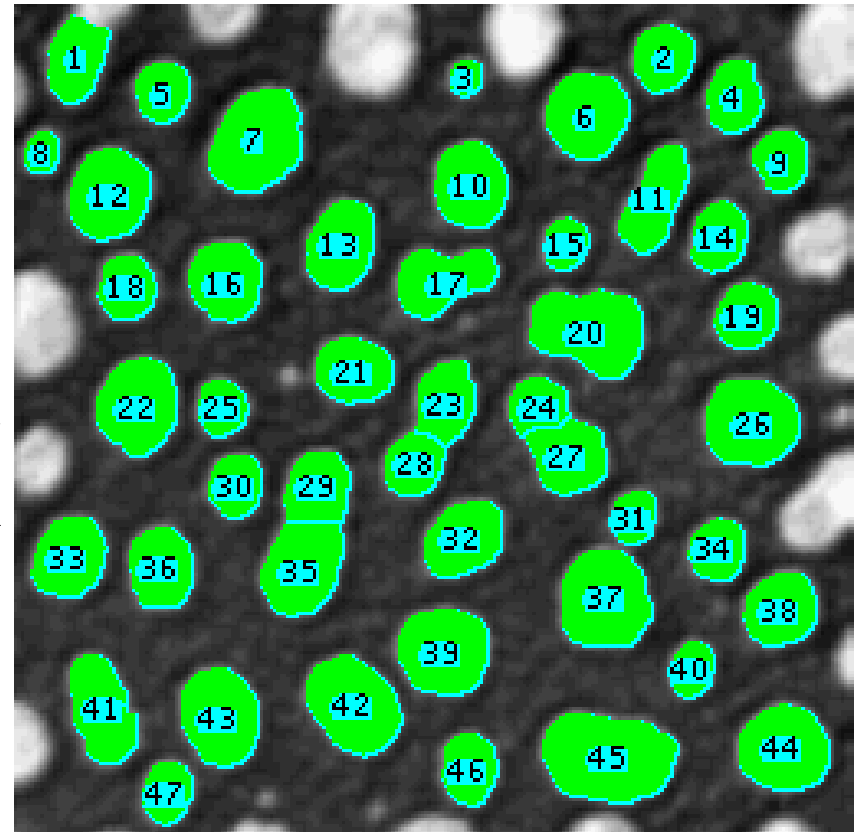
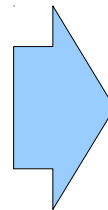
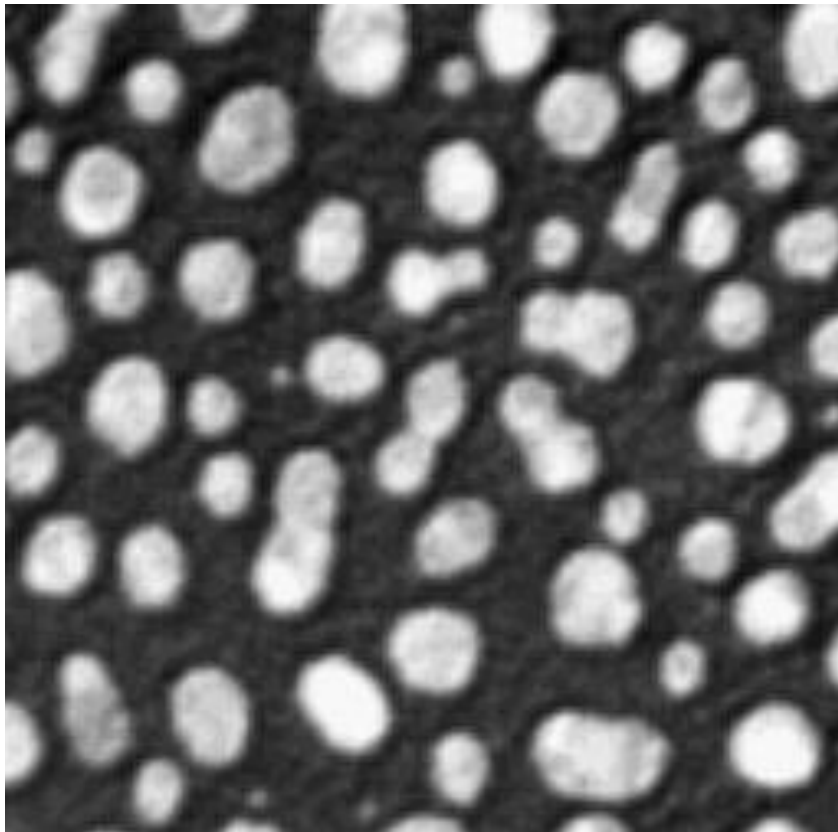
Warum heißt das Breiten-/Tiefen-Suche?



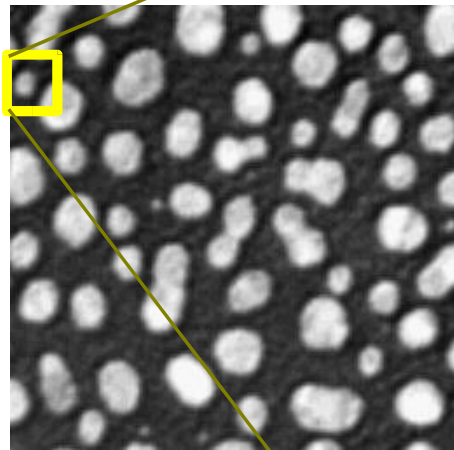
- Für beide Verfahren gilt:
 - Konstante Arbeit für jeden Knoten und jede Kante
 - Die Laufzeit ist also genau $\Theta(|V'| + |E'|)$
wobei V' und E' gerade die Menge aller Knoten und Kanten in der ZK sind, in der der Startknoten liegt
 - Das kann man also (bis auf einen konstanten Faktor) nicht besser machen

Anwendung in der Bildverarbeitung

- Connected Component Labelling
- Zählen von Objekten in einem Bild



Was ist Objekt, was ist Hintergrund?

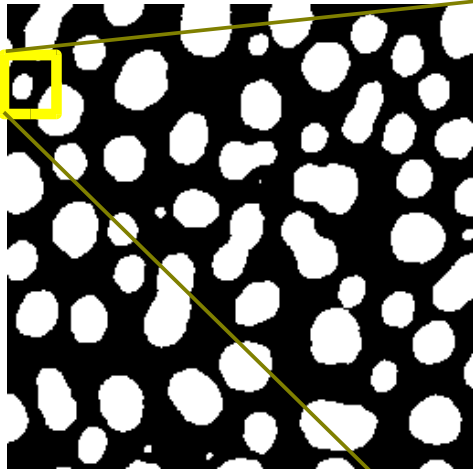


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
35	104	80	56	40	16	16	8	16	16	24	32	32	32	32	32	32	32	32
36	80	64	48	32	16	16	16	24	32	40	40	40	40	40	40	40	32	32
37	56	48	32	24	8	16	16	32	40	48	48	48	40	40	40	40	32	32
38	40	32	24	24	16	32	48	64	72	80	80	72	56	56	48	48	40	40
39	16	16	16	24	24	48	72	88	104	112	112	96	72	64	56	48	40	40
40	16	16	24	40	56	88	120	128	136	144	144	120	96	88	72	56	48	48
41	8	16	24	56	80	120	160	168	168	168	168	144	120	104	80	64	48	48
42	16	32	40	80	112	144	176	176	176	176	168	152	128	112	88	64	48	40
43	24	40	56	96	136	160	184	184	176	176	168	152	136	112	88	64	40	32
44	40	56	80	112	152	168	184	184	176	176	168	152	136	112	80	64	40	32
45	48	72	96	128	160	176	184	184	176	176	168	152	136	104	72	56	32	24
46	48	72	96	136	168	176	192	192	184	184	176	160	136	104	72	56	32	24
47	48	72	96	136	168	184	192	192	192	192	184	160	136	104	72	48	24	24
48	48	72	96	128	168	184	200	200	200	192	184	160	128	96	64	48	24	32
49	48	72	88	128	160	184	200	200	200	192	184	152	120	88	56	40	24	32
50	48	64	80	112	136	160	176	176	176	168	160	136	104	80	48	40	32	40
51	48	64	72	96	112	128	144	152	152	144	136	112	88	64	40	40	32	48
52	48	56	64	80	88	104	112	112	120	112	104	88	72	56	32	32	32	64
53	40	48	48	56	64	72	72	80	80	80	72	64	48	40	24	32	32	72
54	48	48	48	48	48	56	56	56	64	56	56	48	40	32	24	40	48	88
55	48	48	40	40	32	32	32	40	40	40	32	32	24	24	24	40	56	104

Threshold at intensity 100

old intensity: 0-99 ? new intensity: 0

old intensity: 100-255 ? new intensity: 255 (or other non-zero value)



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
35	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	255	255	255	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0
41	0	0	0	0	0	255	255	255	255	255	255	255	255	255	0	0	0	0	0
42	0	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
43	0	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
44	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
45	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
46	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
47	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
48	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
49	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
50	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
51	0	0	0	0	255	255	255	255	255	255	255	255	0	0	0	0	0	0	0
52	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0	0
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255
56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255

what next?

Bild als Graph interpretieren

- jedes weiße Pixel ist ein Knoten
- Kanten zwischen benachbarten Pixeln (üblicherweise 4 oder 8 Nachbarn)
- Kanten werden nicht extra gespeichert, Algorithmus arbeitet direkt auf dem 2D Array
- Breiten- oder Tiefensuche findet dann alle zusammenhängenden Komponenten = Partikel

Find Connected Components

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
35	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	255	255	255	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0
41	0	0	0	0	0	255	255	255	255	255	255	255	255	0	0	0	0	0	0
42	0	0	0	0	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
43	0	0	0	0	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
44	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
45	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
46	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
47	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
48	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
49	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
50	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
51	0	0	0	0	255	255	255	255	255	255	255	255	0	0	0	0	0	0	0
52	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0	0
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	25

- Search pixel-by-pixel for non-zero intensity
- label found pixel as component 1

Find Connected Components

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
35	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	255	255	255	255	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0
41	0	0	0	0	0	255	255	255	255	255	255	255	255	0	0	0	0	0	0
42	0	0	0	0	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
43	0	0	0	0	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
44	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
45	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
46	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
47	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
48	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
49	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
50	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
51	0	0	0	0	255	255	255	255	255	255	255	255	0	0	0	0	0	0	0
52	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0	0
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	25

- Search pixel-by-pixel for non-zero intensity
- label found pixel as component 1
- **check neighbors of all new labeled pixels**

Find Connected Components

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
35	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	255	255	255	255	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0
41	0	0	0	0	0	255	255	255	255	255	255	255	255	0	0	0	0	0	0
42	0	0	0	0	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
43	0	0	0	0	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
44	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
45	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
46	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
47	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
48	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
49	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
50	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
51	0	0	0	0	255	255	255	255	255	255	255	255	0	0	0	0	0	0	0
52	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0	0
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	25

- Search pixel-by-pixel for non-zero intensity
- label found pixel as component 1
- check neighbors of all new labeled pixels
- label non-zero pixels as component 1

Find Connected Components

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
35	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	255	255	255	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0
41	0	0	0	0	0	255	255	255	255	255	255	255	255	0	0	0	0	0	0
42	0	0	0	0	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
43	0	0	0	0	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
44	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
45	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
46	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
47	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
48	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
49	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
50	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
51	0	0	0	0	255	255	255	255	255	255	255	255	0	0	0	0	0	0	0
52	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0	0
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	25	

- Search pixel-by-pixel for non-zero intensity
- label found pixel as component 1
- **check neighbors of all new labeled pixels**
- label non-zero pixels as component 1

Find Connected Components

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
35	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	255	255	255	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0
41	0	0	0	0	0	255	255	255	255	255	255	255	255	255	0	0	0	0	0
42	0	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
43	0	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
44	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
45	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
46	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
47	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
48	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
49	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
50	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
51	0	0	0	0	255	255	255	255	255	255	255	255	0	0	0	0	0	0	0
52	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0	0
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	25

- Search pixel-by-pixel for non-zero intensity
- label found pixel as component 1
- check neighbors of all new labeled pixels
- label non-zero pixels as component 1

Find Connected Components

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
35	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	255	255	255	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0
41	0	0	0	0	0	255	255	255	255	255	255	255	255	255	0	0	0	0	0
42	0	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
43	0	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
44	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
45	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
46	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
47	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
48	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
49	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
50	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
51	0	0	0	0	255	255	255	255	255	255	255	255	0	0	0	0	0	0	0
52	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0	0
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	25

- Search pixel-by-pixel for non-zero intensity
- label found pixel as component 1
- **check neighbors of all new labeled pixels**
- label non-zero pixels as component 1

Find Connected Components

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
35	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	255	255	255	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0
41	0	0	0	0	0	255	255	255	255	255	255	255	255	255	0	0	0	0	0
42	0	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
43	0	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
44	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
45	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
46	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
47	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
48	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
49	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
50	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
51	0	0	0	0	255	255	255	255	255	255	255	255	0	0	0	0	0	0	0
52	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0	0
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	25

- Search pixel-by-pixel for non-zero intensity
- label found pixel as component 1
- check neighbors of all new labeled pixels
- label non-zero pixels as component 1

Find Connected Components

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
35	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	255	255	255	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0	0
41	0	0	0	0	0	255	255	255	255	255	255	255	255	0	0	0	0	0	0
42	0	0	0	0	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
43	0	0	0	0	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
44	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
45	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
46	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
47	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
48	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
49	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
50	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
51	0	0	0	0	255	255	255	255	255	255	255	255	0	0	0	0	0	0	0
52	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0	0
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255

- Search pixel-by-pixel for non-zero intensity
- label found pixel as component 1
- **check neighbors of all new labeled pixels**
- label non-zero pixels as component 1

Find Connected Components

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
35	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	255	255	255	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0
41	0	0	0	0	0	255	255	255	255	255	255	255	255	255	0	0	0	0	0
42	0	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0
43	0	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0
44	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
45	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
46	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
47	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
48	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
49	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
50	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
51	0	0	0	0	255	255	255	255	255	255	255	255	0	0	0	0	0	0	0
52	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0	0
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	25

- Search pixel-by-pixel for non-zero intensity
- label found pixel as component 1
- check neighbors of all new labeled pixels
- label non-zero pixels as component 1

Find Connected Components

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
35	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	255	255	255	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0
41	0	0	0	0	0	255	255	255	255	255	255	255	255	255	0	0	0	0	0
42	0	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
43	0	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
44	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
45	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
46	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
47	0	0	0	255	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0
48	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
49	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
50	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
51	0	0	0	0	255	255	255	255	255	255	255	255	0	0	0	0	0	0	0
52	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0	0
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	255

- Search pixel-by-pixel for non-zero intensity
- label found pixel as component 1
- check neighbors of all new labeled pixels
- label non-zero pixels as component 1

Find Connected Components

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
35	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	255	255	255	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0
41	0	0	0	0	0	255	255	255	255	255	255	255	255	0	0	0	0	0	0
42	0	0	0	0	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
43	0	0	0	0	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
44	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
45	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
46	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
47	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
48	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
49	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
50	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
51	0	0	0	0	255	255	255	255	255	255	255	255	0	0	0	0	0	0	0
52	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0	0
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	25

- Search pixel-by-pixel for non-zero intensity
- label found pixel as component 1
- check neighbors of all new labeled pixels
- label non-zero pixels as component 1

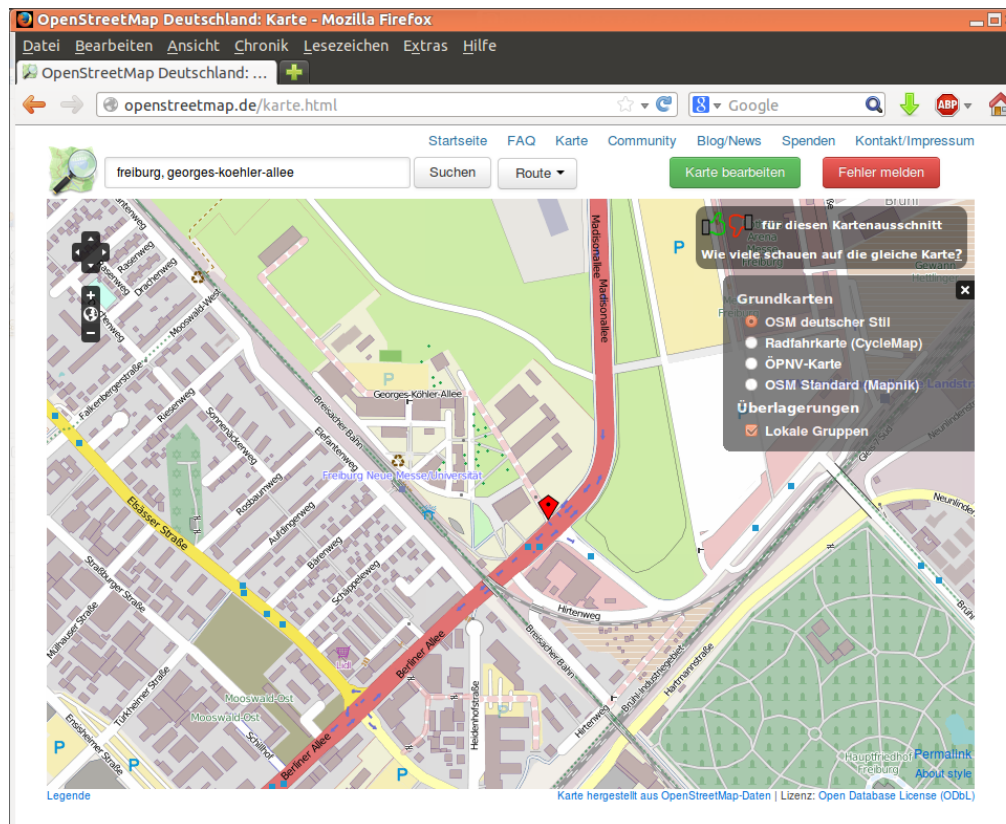
Find Connected Components

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
35	255	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	255	255	255	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0
41	0	0	0	0	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
42	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
43	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
44	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
45	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
46	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
47	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
48	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
49	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
50	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
51	0	0	0	255	255	255	255	255	255	255	255	255	255	0	0	0	0	0	0
52	0	0	0	0	0	255	255	255	255	255	255	0	0	0	0	0	0	0	0
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	25
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	255	25	

- Search pixel-by-pixel for non-zero intensity
- label found pixel as **component 2**
- ...

Übungsblatt

- Größte Zusammenhangskomponente im Saarland aus OpenStreetMap
- <http://openstreetmap.de/karte.html>



- OSM-Graph aufbereitet (nur Straßen) auf Homepage als saarland.graph
- Format (Details siehe „Graph.H“)

284625	Anzahl der Knoten
562361	Anzahl der Kanten
0 49.3414 7.30149	Koordinaten von Knoten 0, latitude 49.3414, longitude 7.30149
1 49.3408 7.30078	Koordinaten von Knoten 1, latitude 49.3408, longitude 7.30078
2 49.3406 7.3003	Koordinaten von Knoten 2, latitude 49.3406, longitude 7.3003
...	...
284624 49.2731 6.70993	Koordinaten von Knoten 284624, latitude 49.2731 , longitude 6.70993
32619 270422 121 130	Kante von Knoten 32619 nach Knoten 270422, Strecke 121 m, max. 130 km/h
32611 1224 90 130	Kante von Knoten 32611 nach Knoten 1224, Strecke 90 m, max. 130 km/h
109 264421 100 30	Kante von Knoten 109 nach Knoten 264421, Strecke 100 m, max. 30 km/h
....	...

Einlesen von Dateien in C++

■ Speziell von zeilenbasierten Daten

– Option 1: **FILE*** und **getline**

Effizient und gut, wenn auch "C Style"

– Option 2: **ifstream** und **getline**

"C++ Style", da gab es früher Probleme mit Dateien > 2GB, aber inzwischen genauso gut wie **FILE*** und **getline**

Felderweise << ist aber ineffizient, erst ganze Zeile lesen !

– Option 3: **fscanf** bzw. **sscanf**

Fehleranfällig und ineffizient, no-no bei großen Daten

– Option 4: **FILE*** und **read**

Alles an einem Stück in einen String einlesen ist natürlich am effizientesten, aber doof wenn nicht alles in den Speicher passt

Literatur / Links

■ Graphen

- In Mehlhorn/Sanders:

8 Graph Representation

- In Wikipedia

[http://en.wikipedia.org/wiki/Graph_\(mathematics\)](http://en.wikipedia.org/wiki/Graph_(mathematics))

■ Graphexploration und Zusammenhangskomponenten

- In Mehlhorn/Sanders:

9 Graph Traversal

- In Wikipedia

http://en.wikipedia.org/wiki/Breadth-first_search

http://en.wikipedia.org/wiki/Depth-first_search

http://en.wikipedia.org/wiki/Connected_component