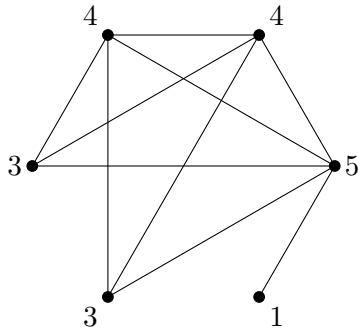


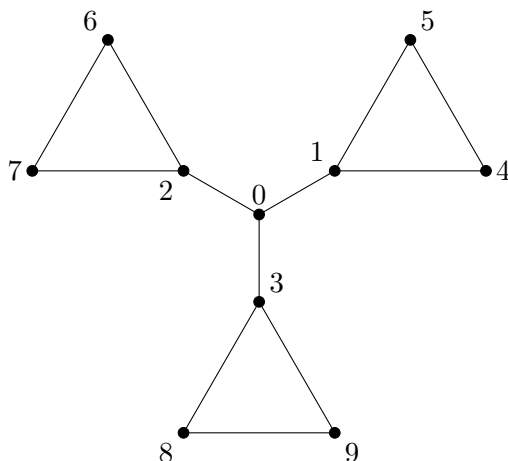
## 7. Übungsblatt

### Aufgabe 1 - Graph Zahl

- Es gibt einen ungerichteten, zusammenhängenden Graph mit mehr als 5 Knoten, der keine Knoten mit demselben Grad hat.



- Das stimmt, da für eine starke Zusammenhangskomponente vorher galt, dass zu jedem Knoten  $v$  in der Zusammenhangskomponente auch einen gerichteten Pfad zu jedem Knoten  $w$  in der Zusammenhangskomponente. Wenn man jetzt die Kantenrichtungen umdreht, dann ist der Weg von einem Knoten  $v$  zu einem Knoten  $w$  der Weg von dem ursprünglichen Graphen von  $w$  nach  $v$ , der aus Voraussetzung existiert. Außerdem bleiben die starken Zusammenhangskomponenten gleich, denn wenn von  $v$  nach  $w$  im ursprünglichen Graphen kein gerichteter Pfad existiert hat, dann existiert im umgedrehten Graphen kein gerichteter Pfad von  $w$  nach  $v$  und umgekehrt.
- Nein nehme folgendes Beispiel:



Wir nennen den Teilgraph mit den Knoten (4) und (5)  $A$ , mit den Knoten (6), (7)  $B$ , mit den Knoten (8), (9)  $C$  und der Teilgraph mit den Knoten (1), (2), (3)  $D$ . Wenn alle Kanten von (1), (2) und (3) zu (0) Richtung (0) Zeigen, dann gibt es keinen gerichteten Pfad zwischen den Teilgraphen  $A$ ,  $B$  und  $C$ .

Wenn zwei der Kanten von  $D$  nach (0) in Richtung (0) zeigen und eine nicht, dann gilt aus Symmetriegründen, dass (E) die Kante von (0) nach (3) zu (3) zeigt. Dann gibt es keinen gerichteten Pfad zwischen den Teilgraphen  $A$  und  $B$ . Wenn nur der Kanten von  $D$  nach (0) in Richtung (0) zeigt, (E ist dies die Kante von (1) nach (0)), dann gibt es keinen gerichteten Pfad zwischen den Teilgraphen  $B$  und  $C$ . Und schlussendlich wenn keine der

Kanten von  $D$  nach  $(0)$  zeigt, dann gibt es wieder keinen gerichteten Pfad zwischen den Teilgraphen  $A, B$  und  $C$

- das stimmt, da wenn alle Zusammenhangskomponenten nur 101 Knoten oder weniger hätten, dann könnte es insgesamt nur weniger als 2020 Knoten geben. (101 in jeder Zusammenhangskomponente)

## Aufgabe 2 - Zusammenhänge erkennen

(a)

**“1.  $\implies$  3.”:** Vollständige Induktion:

**I.A.**  $|V| = 1$  dann ist  $G$  zusammenhängend und hat  $0 = n - 1$  Kanten.

**I.V.** Jeder Baum mit  $|V| = n$  Knoten ist zusammenhängend und hat  $n - 1$  Kanten

**I.S.** Wenn man einen Knoten  $v$  einem Baum hinzufügt, dann muss mindestens eine Kante zum Knoten  $w$  hinzukommen. Würde noch eine weitere Kante zum Knoten  $u$  hinzukommen, dann gäbe es von  $v$  zu  $w$  zwei Wege und zwar der direkte Weg von  $v$  nach  $w$  und der Weg von  $v$  nach  $u$  nach  $w$ . Der Weg von  $u$  nach  $w$  existiert, da wenn er nicht existieren würde  $v$  und  $w$  keine Verbindung hätten und somit in verschiedenen Zusammenhangskomponenten wären.

**“3.  $\implies$  4.”:** Nach der Vorlesung gilt  $m \geq n - k$  mit  $m$  Anzahl der Kanten,  $n$  Anzahl der Knoten und  $k$  Anzahl der Zusammenhangskomponenten. Also Angenommen man kann eine Kante entfernen so, dass es immer noch nur eine Zusammenhangskomponente gäbe, dann würde die Ungleichung  $n - 2 \geq n - 1$  nicht gelten, was zu einem Widerspruch führt und jede Kante die entfernt wird, führt dazu das der Graph unzusammenhängend wird.

**“4.  $\implies$  5.”:** Nehme an es gibt einen Zyklus, dann nehme zwei benachbarte Knoten  $v$  und  $w$  aus Zyklus und entferne die Kante  $(v, w)$ , der Graph wäre immer noch zusammenhängend, da es immer noch einen Weg von  $v$  nach  $w$  gibt, also alle anderen Wege, die über  $(v, w)$  gingen durch den längeren weg über den Zyklus gehen können. Noch zu zeigen maximal Kreisfrei. Durch Kontraposition  $G$  kreisfrei, aber man kann eine Kante zwischen zwei Knoten  $v, w$  hinzufügen, so dass kein Kreis entsteht. Zu zeigen  $G$  ist nicht zusammenhängend. Angenommen  $G$  wäre zusammenhängend, dann gäbe es zwischen  $v$  und  $w$  eine Verbindung. Daraus folgt aber auch, dass die Kante  $(v, w)$  einen Zyklus erzeugen würde. Da man von  $v$  nach  $w$  über die Verbindung in  $G$  und dann von  $w$  direkt zu  $v$  gehen kann.

**“5.  $\implies$  1.”:** Zu zeigen zusammenhängend und kreisfrei. Angenommen nicht zusammenhängend, das heißt es gibt zwei Knoten  $v, w$ , die nicht durch einen Pfad verbunden sind. Also wenn man eine Kante zwischen  $v$  und  $w$  hinzufügt wird, auch kein Kreis erzeugt. Kreisfrei da maximal kreisfrei kreisfrei impliziert.

(b)

**“ $\implies$ ”:** Nach Definition von stark zusammenhängend gibt es zu jedem Knoten  $v$  einen gerichteten Pfad zu jedem  $w$  und von jedem solcher Knoten  $w$  gibt es zu jedem Knoten  $u$  einen gerichteten Pfad, also auch von  $w$  nach  $v$ .

**“ $\Leftarrow$ ”:** Zu zeigen es gibt zu jedem Knoten  $v$  einen Weg zu jedem Knoten  $w$ . Sei  $v$  ein Knoten in  $G(V, E)$ , dann gibt es nach Voraussetzung einen gerichteten Pfad von  $v$  nach  $w$

## Aufgabe 3 - Stabiler Kreislauf

(a) Die Länge ist dann  $v.length$  und der Weg ist dann Die List von vorne nach hinten. Da

---

**Algorithmus 1: BFS()**

---

```
1 let  $Q$  be a queue
2 for all nodes  $w$  with an edge from  $v$  to  $w$  do
3   if  $w = v$  then
4      $w.length = 1$ 
5      $w.parent = v$ 
6     return
7    $Q.enqueue(w)$ 
8 unlable  $v$                                      // If  $v$  was labeled unlabel it
9 while  $Q$  is not empty do
10    $w := Q.dequeue()$ 
11   for all nodes  $u$  with an edge from  $w$  to  $u$  do
12     if  $u$  is unlabeled then
13        $u.length \leftarrow w.length + 1$ 
14        $u.parent \leftarrow w$ 
15       label  $u$  as explored
16        $Q.enqueue(u)$ 
17   if  $u = v$  then
18     return
```

---

---

```
1  $w \leftarrow v.parent$ 
2 let  $List$  be a list
3  $List.append(v)$ 
4 while  $w$  is not  $v$  do
5    $List.append(w)$ 
6    $w \leftarrow w.parent$ 
7  $List.append(v)$ 
8 turn  $List$  around so that the first element is the last and the last is the first and vice versa
9 return  $List$ 
```

---

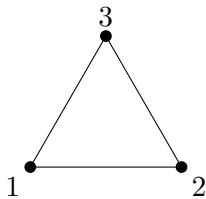
der erste Algorithmus jeden Knoten nur höchstens einmal besucht, und jede Kante so nur höchstens einmal abfragt ist die Laufzeit in  $|\mathcal{V}| + |\mathcal{E}|$ .

Der Algorithmus bedient sich dem Breitensuche-Algorithmus um den kürzesten Weg von zwei Knoten zu finden, wobei hier der Anfangspunkt gleich dem Endpunkt ist, sodass der Anfangspunkt wieder unlabeled wurde, sodass der kürzeste Weg von  $v$  nach  $v$  gesucht wurde.

- (b) Wenn bei der Breitensuche man einen Knoten findet, der schon gelabeled ist, dann gibt es einen Zyklus, da der Knoten von der Wurzel durch zwei verschiedene Wege erreicht werden kann, also wäre ein möglicher Zyklus von der Wurzel über den einen Weg zum schon gelabelten Knoten zurück über den anderen Weg zur Wurzel.
- (c) Nach Aufgabe 2 ist azyklisch äquivalent dazu dass der Graph  $n - 1$  Kanten hat. Wenn weniger als zwei Knoten einen Grad von  $\leq 1$  haben, dann ist die Summer aller Grade größer als  $2(n-1) + 1 = 2n - 1$ . Gäbe es  $n - 1$  Kanten in diesem Graphen, dann ergibt dies aber die Summe aller Grade von  $2(n - 1)$ , da eine Kante den Grad von zwei Knoten um eins erhöht und wenn es null Kanten gibt, dann ist auch der Grad aller Knoten zusammengezählt 0. Da aber  $2(n - 1) = 2n - 2 < 2n - 1$  und die Summer aller Grade größer ist als  $2n - 1$ , führt dies zu einem Widerspruch.

#### Aufgabe 4 - Zwei Farben

(a)



- (b) Worst case alle Knoten und alle Kanten  $\mathcal{O}(|V| + |E|)$

---

```

1 for  $v$  in Knoten do
2   for  $w$  Nachbar von  $v$  do
3     if  $v.color = w.color$  then
4       return nicht zweifarbig
5 return zweifarbig

```

---

- (c) Siehe b)?