# Übungsblatt 06

## Algorithmen I - Sommersemester 2024

**Abgabe im ILIAS bis 07.06.2024, 18:00 Uhr**

Die Abgabe erfolgt alleine oder zu zweit als *eine* PDF-Datei über das Übungsmodul in der Gruppe eures Tutoriums im ILIAS. Beschriftet die Abgabe deutlich mit Matrikelnummer und Name.

- Achtet bei handschriftlichen Abgaben auf Lesbarkeit.
- Achtet darauf, ob Algorithmen in Worten oder Pseudocode beschrieben werden sollen.
- Es werden immer asymptotisch möglichst effiziente Algorithmen erwartet, wenn nicht anders angegeben.
- Wenn Korrektheits- oder Laufzeitanalysen gefordert sind, behandelt diese separat von der Algorithmenbeschreibung.

**Gesamtpunkte**: 20 (+ 2 Bonus)

## Aufgabe 1 - Weil Baum! (7 Punkte)

Da Bäume wichtige und grundlegende Graphen sind, sollst du in dieser Aufgabe Bäume und ein paar ihrer Eigenschaften näher kennen lernen. Wir beschränken uns dabei auf Graphen mit mindestens einem Knoten.

a) Zeige, dass jeder Baum mindestens ein Blatt, also einen Knoten mit Grad höchstens 1, enthält. (1.5 Punkte)

b) Sei $T$ ein Graph mit $n$ Knoten. Gegeben sind die folgenden drei Definitionen eines Baumes:

   1) $T$ ist nach der Definition aus der VL ein Baum, also: $T$ ist zusammenhängend und kreisfrei,
   2) $T$ ist kreisfrei und hat $n-1$ Kanten,
   3) $T$ ist zusammenhängend und hat $n-1$ Kanten.

   Zeige, dass die drei Definitionen äquivalent sind. (5.5 Punkte)

## **Aufgabe 2 - Die Musterlösung hat heute eine Verspätung von ca. 10 Tagen** (8 Punkte)

Um seinen Geheimplan umsetzen zu können, muss Dr. Meta im nächsten Schritt seine Biber an strategisch wichtigen Punkten in ganz Deutschland verteilen. Dafür möchte er mithilfe des Deutschlandtickets seine Biber mit dem Zug reisen lassen. Beim Besuch der Vorlesung Algorithmen I hat Dr. Meta gehört, dass man mithilfe von Breitensuche kürzeste Wege berechnen kann. Allerdings kam ihm just in diesem Moment eine Idee, wie er das [kœri]-Werk stilllegen kann und daher hat er die genauen Details nicht mitbekommen. Aufgrund dessen beauftragt er seine besten Biber mit der Erforschung des Themas.

Diese modellieren das Schienennetz als zusammenhängende und ungerichtete Graphen. In der Vorlesung haben wir gesehen, dass Breitensuche in ungewichteten Graphen kürzeste Pfade von einem Startknoten $s$ zu allen anderen Knoten findet. In den ersten beiden Teilaufgaben wollen wir uns nochmal *ungewichtete* Graphen genauer anschauen.

a) Wir repräsentieren die Knoten eines Graphen $G$ durch die Zahlen 0 bis $n-1$, wobei $n$ die Anzahl der Knoten ist. Nun wollen wir das Array bestimmen, das für jeden Index $t$ die Distanz $\text{dist}(s,t)$ von Startknoten $s$ zu Knoten $t$ enthält. Beschreibe, wie du die Breitensuche anpassen musst, um dieses Array berechnen zu können. Die Laufzeit der Breitensuche soll sich dabei nicht ändern. (2 Punkte)

b) Nun wollen wir nicht nur die Distanz von $s$ zu einem Knoten $t$ bestimmen, sondern zusätzlich auch den kürzesten Pfad. Beschreibe, welche Informationen man sich zusätzlich zum Distanzarray noch während der BFS speichert und wie man damit dann den kürzesten Pfad von $s$ zu $t$ rekonstruiert. Auch hier soll sich die Laufzeit der Breitensuche nicht ändern. (2 Punkte)

Bei der Modellierung des Schienennetzes haben die Biber allerdings die Fahrtdauer in Stunden auf einer Strecke als ganzzahliges und positives Kantengewicht ($> 0$) verwendet. Wir betrachten also nun *gewichtete* Graphen.

c) Gib einen Beispielgraphen mit Kantengewichten und Knoten $s$ und $t$ an, bei dem die Breitensuche aus der Vorlesung nicht den kürzesten Pfad zwischen $s$ und $t$ findet. Begründe. (0.5 Punkte)

Auf der Suche nach einem Ausweg aus der Klemme, steckt dir die Agentin Johnson, die Dr. Meta in den Bahnvorstand eingeschleust hat, folgende brisante Information zu: Im Schienennetz der Deutschen Bahn braucht ein Zug von einem Bahnhof zum nächsten höchstens $k$ Stunden.

d) Beschreibe nun einen Algorithmus, der mithilfe von Breitensuche in einem Graphen, in dem jede Kante höchstens Gewicht $k$ hat, die kürzesten Pfade vom Startknoten $s$ zu allen anderen Knoten bestimmt. (2 Punkte)
*Tipp*: Modifiziere den Graphen so, dass die Breitensuche das korrekte Ergebnis liefert.

e) Da die Zeit drängt, musst du Dr. Meta zeigen, dass dein Algorithmus ausreichend schnell ist. Gib hierfür die Laufzeit deines Algorithmus in Abhängigkeit von $k$, Knotenzahl $n$ und Kantenzahl $m$ an. Begründe diese. (1 Punkt)

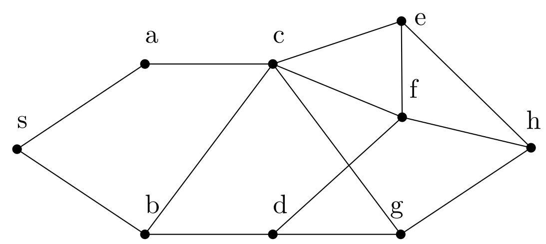
f) Wann stößt dein Algorithmus auf Probleme und wird auf auch auf gewichteten Graphen mit wenigen Knoten bzw. Kanten zu langsam? (0.5 Punkte)

Es meldet sich bei dir ein mysteriöser Agent mit dem Decknamen Dijkstra. *To be continued...*

## Aufgabe 3 - Wie viele Wege führen nach Rom? (5 Punkte)

Gegeben sei ein ungerichteter und ungewichteter Graph. Wir wollen uns in dieser Aufgabe damit beschäftigen, wie viele kürzeste Pfade es jeweils zwischen zwei Knoten gibt. Zwei Pfade sind verschieden, wenn sie sich in mindestens einem Knoten unterscheiden.

a) Gegeben sei der folgende Beispielgraph. Gib für jeden der Knoten $a, b, c, d, e, f, g, h$ an, wie viele kürzeste Wege es zwischen $s$ und diesem Knoten gibt. (1 Punkt)

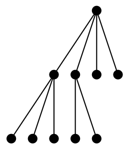



b) Beschreibe eine Breitensuchvariante, die bestimmt, wie viele kürzeste Pfade es zwischen einem gegebenen Knoten $s$ und jedem anderen Knoten gibt. Gib außerdem die Laufzeit des Algorithmus abhängig von Knotenzahl $n$ und Kantenzahl $m$ an und begründe diese. Begründe auch die Korrektheit deines Ansatzes. (4 Punkte)

## **Aufgabe 4 - Zum Knobeln** (0 Punkte + 2 Bonus)

Dr. Meta befürchtet, dass seine Biber vor lauter Bäumen den Wald nicht mehr sehen. Daher sollst du deinen guten Willen unter Beweis stellen und den Bibern einige Bäume zeigen. Dr. Meta wünscht sich, dass die von dir gezeichneten Bäume klein sind. Daher hat er sich auf eine Knotenzahl von **genau** 10 festgelegt. Außerdem mag er die Zahl zwei nicht, daher möchte er in deinen Beispielen keine Knoten mit Grad zwei finden. Dr. Metas finaler Wunsch ist, dass die Bäume auch wirklich verschieden, also nicht isomorph[1] sind.

Zeichne alle Bäume, die Dr. Metas Spezifikation erfüllen. Gib zusätzlich für jeden Baum seine Gradsequenz in absteigender Reihenfolge an.



$[4, 4, 3, 1, 1, 1, 1, 1, 1, 1]$

Beispiel: für einen Baum, der die Spezifikation erfüllt mit seiner Gradsequenz. Der Baum enthält zwei Knoten mit Grad 4, einen Knoten mit Grad 3 und sieben Knoten mit Grad 1.

(2 Punkte)

*Hinweis:* Die Gesamtanzahl der Bäume ist ausreichend klein, dass du diese problemlos zeichnen kannst.

[1] Zwei Graphen $G = (V_G, E_G)$ und $H = (V_H, E_H)$ sind isomorph, falls sie die gleiche Struktur haben. Formal sind $G$ und $H$ isomorph, falls es eine Abbildung $f : V_G \to V_H$ gibt, sodass $\forall u, v \in V$ gilt: $\{u, v\} \in E_G$ gdw. $\{f(u), f(v)\} \in E_H$.