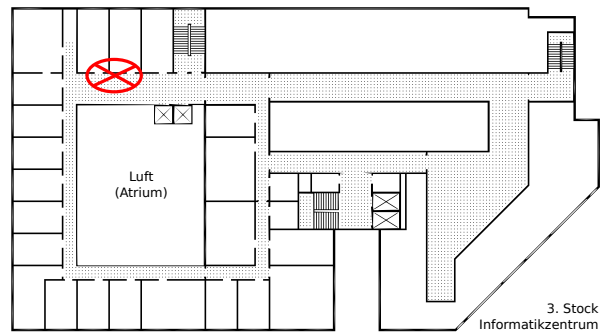


Prof. Dr. Sándor P. Fekete  
Christian Rieck  
Arne Schmidt

## Algorithmen und Datenstrukturen Übung 2 vom 20. 11. 2017

Abgabe der Lösungen bis zum Montag, den 04.12.2017 um 11:30 im Hausaufgabenschrank bei Raum IZ 337. Es werden nur mit einem dokumentenechten Stift (kein Rot!) geschriebene Lösungen gewertet. **Bitte die Blätter zusammenheften und vorne deutlich mit eigenem Namen, Matrikel- und Gruppennummer, sowie Studiengang versehen!**



**Aufgabe 1 (Breiten- und Tiefensuche):**  
Betrachte den Graphen  $G$  in Abbildung 1.

(10+10+5 Punkte)

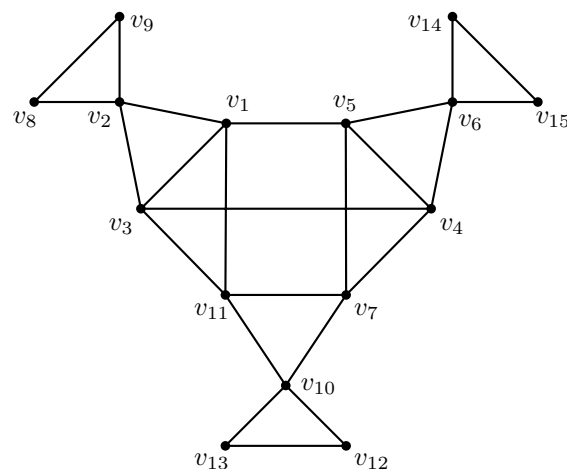


Abbildung 1: Der Graph  $G$ .

- Wende Breitensuche auf  $G$  mit Startknoten  $v_1$  an. Gib dabei **jede** Änderung von  $R$  an. Gib den gefundenen Baum an. Kommen zu einem Zeitpunkt mehrere Knoten in Frage, wähle den Knoten mit kleinstem Index.
- Wende Tiefensuche auf  $G$  mit Startknoten  $v_1$  an. Gib dabei **jede** Änderung von  $R$  an. Gib den gefundenen Baum an. Kommen zu einem Zeitpunkt mehrere Knoten in Frage, wähle den Knoten mit kleinstem Index.
- Gib die Adjazenzliste von  $G$  an.

**Aufgabe 2 (Tiefensuche):****(5+5+5 Punkte)**

Sei  $B_G$  der Tiefensuchbaum von einem beliebigen Graphen  $G$  und sei  $u$  der Startknoten der Tiefensuche.

- Zeige: Wenn  $u$  in  $B_G$  einen Grad von mindestens zwei besitzt, dann zerfällt  $G$  nach Entfernen von  $u$  aus  $G$  (mitsamt inzidenter Kanten) in mindestens zwei Zusammenhangskomponenten.
- Zeige, dass auch die Umkehrung von a) gilt: Wenn  $G$  nach Entfernen von  $u$  (mitsamt inzidenter Kanten) in mindestens zwei Zusammenhangskomponenten zerfällt, besitzt  $u$  in  $B_G$  einen Grad von mindestens zwei.
- Gilt die Aussage in a) auch für jeden anderen Knoten in  $B_G$ ? Begründe deine Antwort!

**Aufgabe 3 (Oracle of Kevin Bacon):****(6+7+7 Punkte)**

Dem *Orakel von Kevin Bacon* liegt der Schauspielergraph  $S$  zugrunde: Schauspieler sind durch Knoten repräsentiert. Zwei Schauspielerknoten sind durch eine Kante verbunden, wenn sie gemeinsam in einem Film gespielt haben. Der Knoten von Kevin Bacon hat die *Kevin-Bacon-Zahl* (KBZ) 0; die KBZ eines anderen Schauspielers ist die Länge eines kürzesten Weges im Schauspielergraphen  $S$  zu Kevin Bacon. Beispielsweise hat Tom Hanks die KBZ 1, da er mit Kevin Bacon in *Apollo 13* gespielt hat. Falls kein verbindender Weg existiert, ist die KBZ des betrachteten Schauspielers als unendlich definiert.

Das Orakel ist im Web verfügbar: <http://oracleofbacon.org/>. Die zugrundeliegenden Filmdata sind der *Internet Movie Database* entnommen: <http://www.imdb.com/>.

- Wir betrachten einen Pfad für die KBZ  $z$  eines Schauspielers  $A$  als ein kürzester Pfad in  $S$ , der Kevin Bacon und  $A$  verbindet und der aus  $z + 1$  Knoten besteht. Gib einen Schauspieler mit mindestens KBZ 4 und einen entsprechenden Pfad an.
- Wenn man Schauspieler mit möglichst großer KBZ sucht, muss man davon ausgehen, dass man diese nicht kennt; man muss sie also erst als Teil der Suche finden. Beschreiben Sie eine allgemeine Strategie zum sicheren Finden (ohne Raten) von Schauspielern mit hoher KBZ, die jeweils nur ein Browserfenster/-tab mit dem Orakel of Bacon und ein Browserfenster/-tab mit der IMDb verwendet. Welche Rolle spielt dabei die Breitensuche? Welche spielt die Tiefensuche?
- Wenn man einen Server wie das Kevin-Bacon-Orakel betreibt, muss man damit rechnen, dass in kurzer Zeit sehr viele Anfragen hereinkommen, die jeweils schnell beantwortet werden müssen. Deshalb lohnt es sich, zwischen Verfahren zu unterscheiden, die eine Aufgabenstellung nur einmal lösen, und solchen, die (nach einem gewissen Aufwand für "Preprocessing" zur Erstellung einer geeigneten Datenstruktur) dieselbe Frage immer wieder neu für unterschiedliche Anfragen ("Queries") beantworten können. Bei einer Anfrage soll nun nicht nur die KBZ des betrachteten Schauspielers, sondern auch ein entsprechender Pfad ausgegeben werden, siehe Aufgabenteil a). Wie kann man es als Betreiber eines Orakels vermeiden, dass man für jede Anfrage eine neue Breitensuche ausführen muss, ohne dass man gigantische Datenmengen vorhalten muss?<sup>1</sup>

---

<sup>1</sup>Hier geht es nicht um rein technische Lösungen wie Caching, sondern um das Ausnutzen von Struktur.