**Universität des Saarlandes**
**FR 6.2 Informatik**

**Prof. Dr. Kurt Mehlhorn** **WiSe 2015/2016**

# Übungen zu Ideen der Informatik

http://www.mpi-inf.mpg.de/departments/algorithms-complexity/teaching/winter15/ideen/

Beispielklausur Abgabeschluss: 04.01.2016

*Bitte beachten Sie, dass der Umfang dieser Beispielklausur nicht dem Umfang der End- und Nachklausur entspricht. Diese Beispielklausur bemüht sich darum, in der Art und Weise der Aufgabenstellungen, ähnlich zur End- und Nachklausur zu sein. Selbstverständlich sind Themen, die in den kommenden Wochen noch in der Vorlesung behandelt werden, genauso für die End- und Nachklausur relevant, wie alle bereits behandelten.*

**Aufgabe 1** (*10 Punkte*)

a) Erklären Sie das "Teile und Herrsche - Prinzip" anhand von Binärsuche. (3 Punkte)

b) Erörtern Sie die Voraussetzung für die Anwendbarkeit von Binärsuche und nennen Sie die Laufzeit des Verfahrens. (2 Punkte)

c) Demonstrieren Sie den Algorithmus anhand des folgenden Beispiels

$$[\text{Neon}, \text{Argon}, \text{Helium}, \text{Xenon}, \text{Radon}, \text{Krypton}]$$

indem Sie das einzige radioaktive Edelgas (Radon) suchen. (Denken Sie daran nötigenfalls die Voraussetzung aus b) zu schaffen.) Illustrieren Sie sämtliche Schritte! (3 Punkte)

d) Welchen Aufwand müssen Sie im Allgemeinen betreiben um die Voraussetzung aus b.) zu schaffen? (2 Punkte)

**Aufgabe 2** (*5 Punkte*)

a) Nennen Sie eine Methode zur von Lösung linearer Optimierungs Problemen. (1 Punkt)

b) Auf welchen Graphen funktioniert Dijkstra's Algorithmus und was muss man bei den Kantengewichten beachten? (1 Punkt)

c) Nennen Sie ein asymmetrisches Verschlüsselungsverfahren. (1 Punkt)

d) Nennen Sie eine Errungenschaft von Alan Turing. (1 Punkt)

e) Mit welchem Algorithmus kann man die Wichtigkeit von Webseiten berechnen? (1 Punkt)

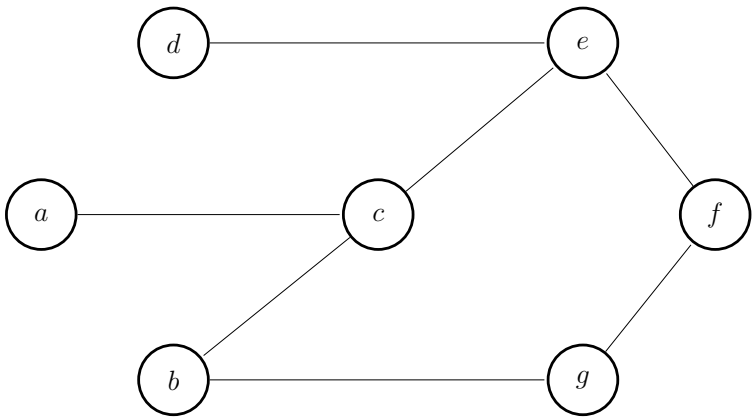
**Aufgabe 3** (*10 Punkte*) Der folgende Algorithmus nennt sich Breitensuche. Als Eingabe dient ein Startknoten $s$ und ein ungerichteter Graph $G = (V, E)$, wobei $V$ die Knoten- und $E$ die Kantenmenge bezeichnen.

**Algorithmus 1 :** Breitensuche

```
   Eingabe : G = (V, E) sowie ein Startknoten s
 1 färbe alle Knoten weiß
 2 färbe s schwarz
 3 L = eine leere Liste
 4 für jeden Nachbarn w von s tue
 5 |  hänge (s, w) ans Ende von L an.
 6 solange L nicht leer ist tue
 7 |  Sei (u, v) der erste Eintrag in L
 8 |  wenn v weiß ist dann
 9 |  |  färbe v schwarz
10 |  |  für jeden Nachbarn w von v tue
11 |  |  |  Hänge (v, w) an L an.
12 |  Lösche den ersten Eintrag aus L
```

a) Führen Sie den Algorithmus auf dem folgenden Graphen aus. Beginnen Sie am Knoten $a$. Wenn Sie über die Nachbarn eines Knoten iterieren, so tun Sie dies immer in alphabethischer Reihenfolge.

Markieren Sie in der Abbildung die schwarzen Knoten und die Reihenfolge, in der sie schwarz gefärbt wurden. (*5 Punkte*)




b) Überlegen Sie sich eine Abschätzung an den Aufwand des Algorithmus abhängig von $n = |V|$ und $m = |E|$. Überlegen Sie sich dazu, wie oft eine Kante zu $L$ hinzugefügt werden kann und wieviel Aufwand der Algorithmus für jede Kante in $L$ betreibt. (*5 Punkte*)