

# Algorithmen und Datenstrukturen Klausur WS 2006/07

## Software-Engineering und Technische Informatik Bachelor

Die Klausur besteht aus 6 Aufgaben und umfasst 60 Punkte.

Bitte schreiben Sie die Lösungen auf die Aufgabenblätter.

Vergessen Sie nicht, Ihren Namen auf die erste Seite einzutragen und versehen Sie zusätzliche Blätter ebenfalls mit Ihrem Namen.

Als Hilfsmittel sind alle schriftlichen Unterlagen gestattet.

Viel Erfolg!

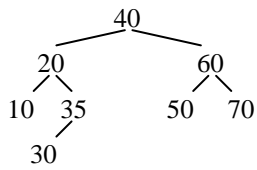
Name	
Studiengang	
Matrikelnummer	

Aufgabe 1	AVL-Baum	12	
Aufgabe 2	B-Baum	10	
Aufgabe 3	Hashverfahren	8	
Aufgabe 4	Tiefen- und Breitensuche	10	
Aufgabe 5	Kürzeste Wege	10	
Aufgabe 6	Flüsse in Netzwerke	10	
Summe		60	

## Aufgabe 1 AVL-Baum

(12 Punkte)

a) Zeigen Sie, dass folgender binärer Suchbaum die AVL-Eigenschaft besitzt.

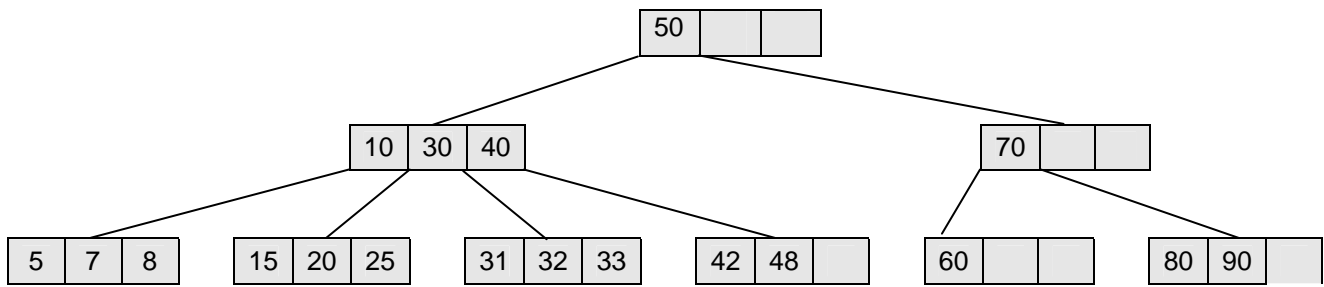


b) Fügen Sie in dem AVL-Baum die Zahlen **25**, **36**, **34**, **37** ein und löschen Sie dann die Zahlen **35** und **37**. Geben Sie an, welche Rotationsoperationen durchgeführt werden.

## Aufgabe 2 B-Baum

(10 Punkte)

Folgender B-Baum der Ordnung  $m = 4$  sei gegeben.



Fügen Sie in den B-Baum die Zahlen **45 und 9** ein und löschen Sie dann die Zahlen **60 und 70**. Geben Sie die jeweils durchgeführten Operationen an.

### Aufgabe 3 Hashverfahren

(8 Punkte)

- a) Eine Menge von deutschen Verben (gehen, springen, lachen, etc.) soll mit einem Hashverfahren abgespeichert werden. Es wird die in der Vorlesung besprochene Hashfunktion für Strings verwendet (siehe Seite 1-26). Welche Problem entsteht, wenn als Tabellengröße  $n = 16348 = 128^2$  gewählt wird? Wie lässt sich das Problem beheben?
- b) Zur Verwaltung von Produkten mit einem Hashverfahren soll die EAN-Nummer (European Article Number) als Schlüssel gewählt werden. EAN-Nummern seien durch folgenden Strukturdatentyp definiert:

```
struct EAN
{
    int ln; // Laender-Nummer; 2- bis 3-stellig
    int hn; // Hersteller-Nummer; 4- bis 5-stellig
    int tn; // Artikelnummer; 5-stellig
};
```

Definieren Sie eine geeignete Hashfunktion

```
int h(EAN ean)
```

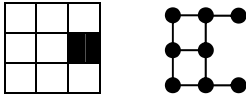
ohne die EAN-Nummer in einen String umzuwandeln.

#### Aufgabe 4 Tiefen- und Breitensuche in einer Hinderniswelt

(10 Punkte)

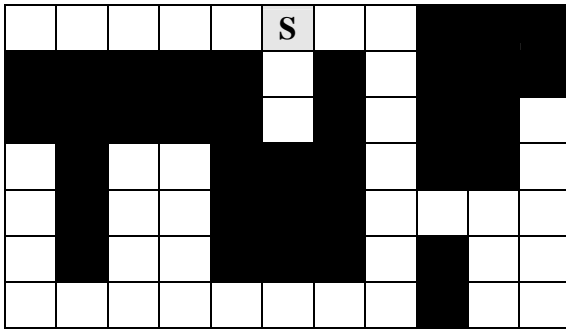
Eine Hinderniswelt lässt sich durch ein Gitter beschreiben, wobei betretbare Zellen weiß und nicht betretbare Zellen schwarz dargestellt werden. Zwei betretbare Zellen sind benachbart, falls sie sich entweder direkt übereinander oder nebeneinander befinden. Damit lässt sich eine Hinderniswelt auch als Graph auffassen, wobei die Knoten gerade die betretbaren Zellen sind.

In der folgenden Abbildung ist eine kleine Hinderniswelt mit dem entsprechenden Graphen zu sehen.

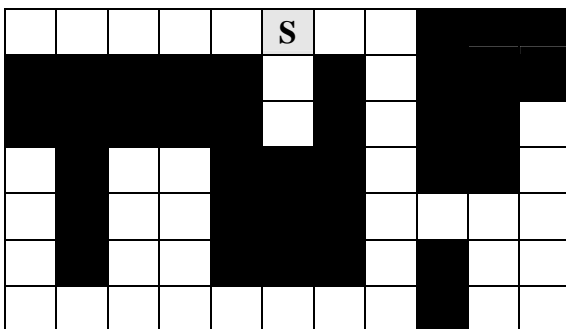


Folgende Hinderniswelt wird bei S beginnend mit Tiefensuche (rekursive Variante) bzw. Breitensuche durchlaufen. Tragen Sie in den Zellen die Besuchszeitpunkte an. Gehen Sie davon aus, dass für eine Zelle die benachbarten Zellen immer in der Reihenfolge N, O, S, W (oberhalb, rechts, unterhalb, links) betrachtet werden.

#### Tiefensuche:



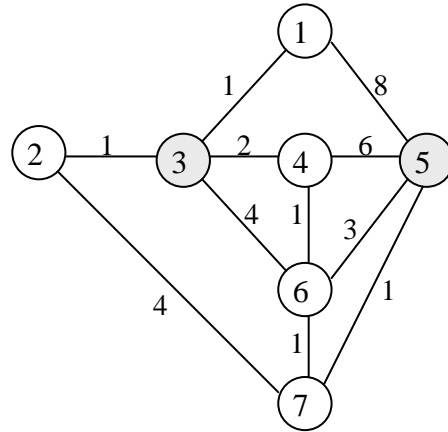
#### Breitensuche:



### Aufgabe 5 Kürzeste Wege in Graphen

(10 Punkte)

Bestimmen Sie mit dem Algorithmus von Dijkstra den kürzesten Weg von Knoten 3 nach 5. Tragen Sie nach jedem Besuchsschritt für jeden Knoten die aktuelle Distanz  $d$  zum Startknoten in die gegebene Tabelle ein. Kennzeichnen Sie außerdem die Kandidaten, die als nächstes besucht werden können. Geben Sie den gefundenen kürzesten Weg an.

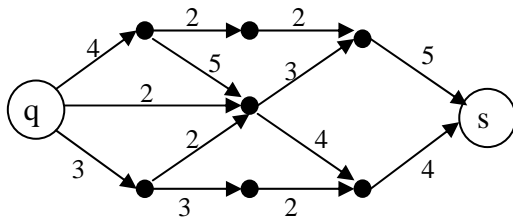


Besucher Knoten	d[1]	d[2]	d[3]	d[4]	d[5]	d[6]	d[7]

### Aufgabe 6 Flüsse in Netzwerke

(10 Punkte)

Im folgenden Graphen ist jede Kante mit ihrer Kapazität markiert. Bestimmen Sie mit dem Algorithmus von Ford-Fulkerson einen maximalen Fluss von der Quelle q zur Senke s. Geben Sie an, mit welcher Methode Sie eine Flussenerweiterung im Residualgraphen suchen.



Aktueller Fluss

Residualgraph:

