Name, Vorname: **Matrikelnummer**:

Klausur zur Vorlesung 12. Februar 2015

Algorithmen und Programmierung 3, WS 2014/15

Helmut Alt

Beginn: 14 Uhr, Ende: 16 Uhr (120 min.)

1.	2.	3.	4.	Form	Σ
/12	/12	/12	/12	/2	/50

◯ Ich bin damit einverstanden, dass mein Klausurergebnis unter Angabe der Matrikelnummer auf einer Webseite veröffentlicht wird.

Als Hilfsmittel ist neben Schreibutensilien ausschließlich ein (doppelseitig) beschriebenes oder bedrucktes Blatt pro Person für den eigenen Gebrauch zugelassen.

Bitte nicht nach zusätzlichem Papier fragen außer im unwahrscheinlichen Fall, dass all diese Blätter voll beschriftet sind. Bitte kein eigenes Schmierpapier benutzen.

Alle Antworten müssen **begründet** werden. Bei allen Algorithmen soll die Funktionsweise erkennbar sein und muss ggf. erklärt werden. Wenn nicht anders angegeben sollen Algorithmen bzgl. ihrer Laufzeit analysiert werden und falls nötig soll auch deren Korrektheit erklärt werden. Ist nach einem *effizienten* Algorithmus gefragt, so kann auch für einen Teil der Punkte ein weniger effizienter angegeben werden. Ergebnisse und Algorithmen aus der Vorlesung dürfen benutzt werden.

Die 2 Punkte für "Form" werden vergeben für Leserlichkeit, korrektes Ausfüllen der persönlichen Daten, u.ä.

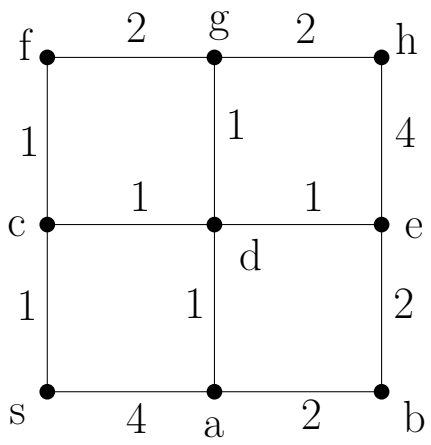
Aufgabe 1 /12

Finden Sie für den folgenden ungerichteten Graphen mit Kantengewichten durch Anwendung von Algorithmen aus der Vorlesung:

(a) die kürzesten Wege vom Knoten s zu allen anderen Knoten. Wieso kann man hier einen Algorithmus für gerichtete Graphen auf ungerichtete übertragen?

(b) einen minimalen Spannbaum.

Geben Sie die Vorgehensweise der Algorithmen Schritt für Schritt an.



Aufgabe 2 /12

Fügen Sie die Wörter DAVE, DEE, DOZY, BEAKY, MICK, TICH, YOU für jede der folgenden Datenstrukturen der Reihe nach in eine zunächst leere Struktur ein. Falls eine Ordnungsrelation nötig ist, wählen Sie die lexikographische Ordnung. Streichen Sie dann das Wort DEE. Zeigen Sie alle Zwischenschritte.

(a) Trie (kompakt, dh. Kanten können mit Strings beschriftet sein)

(b) Hashtabelle der Größe 5 wobei Kollisionen durch Verkettung aufgelöst werden. Dabei soll folgende Hashfunktion verwendet werden: Jedem Buchstaben wird ein Wert gemäß seiner Stellung im Alphabet zugeordnet also

A	B	C	D	E	H	I	K	M	O	T	U	V	Y	Z
1	2	3	4	5	8	9	11	13	15	20	21	22	25	26

Für ein Wort werden die Werte der einzelnen Buchstaben addiert und modulo 5 genommen.

(c) AVL-Baum

Aufgabe 3 /12

Welche der folgenden Entscheidungsprobleme liegen in P? Geben Sie in diesem Fall möglichst effiziente Algorithmen an. Welche liegen in NP und für welche lässt sich NP-Vollständigkeit zeigen?

(a) Gegeben eine Folge von n Zahlen, stelle fest, ob alle Zahlen verschieden sind.

(b) Gegeben eine Folge von n ganzen Zahlen, stelle fest, ob es drei davon gibt, die sich zu 0 addieren. *Anmerkung:* $O(n^3)$ ist möglich aber nicht optimal.

(c) Gegeben 2 Wörter $u, v \in \{0,1\}^*$, beide der Länge n, stelle fest, ob es zwei gemeinsame Teilwörter der Länge 5 von u und v gibt, die sich nicht überlappen.

(d) Gegeben eine Folge von n Gegenständen mit Volumina $v_1, ..., v_n \in \mathbb{N}$ und Zahlen $k, v \in \mathbb{N}$, stelle fest, ob sich die Gegenstände in k Kisten, die alle das Volumen v haben, packen lassen. *Hinweis:* SUBSET-SUM oder Aufteilung einer Zahlenfolge in zwei Hälften mit gleicher Summe.

Aufgabe 4 /12

Schreiben Sie in JAVA Klassen für

(a) binäre Suchbäume

(b) 2-3-Bäume

Beide Klassen sollen jeweils die folgenden beiden Schnittstellen implementieren, und zwar nur die dort angegebenen Methoden. `einfuege()` braucht nicht implementiert zu werden.

```
public interface Woerterbuch<S extends Comparable<S>,W> {
  public W finde(S schluessel);
  public void einfuege(S schluessel, W wert);
  public boolean istLeer();
}


public interface Baum<E> {
  public List<E> wurzelInhalte();        //Liste der in der Wurzel
                                         //gespeicherten Objekte

  public List<Baum<E>> unterBaeume();    // Unterbäume der Wurzel
}
```