

Name:

Matrikelnummer:

PROBEKLAUSUR

Informatik B

Februar 2003

Teil I: Informatik 3

Seite 1 von 5

ACHTUNG

Die vorliegende Probeklausur umfaßt lediglich den **Informatik 3 Teil** der Informatik B Klausur und ist für eine Bearbeitungszeit von etwa **50 Minuten** gedacht. In der realen Klausur stehen für beide Teile (Info3 und Info4) insgesamt 240 Minuten zur Verfügung!

Dieses Muster soll den *möglichen Aufbau* und die *Art der Fragestellung* des Info3-Teils der Klausur beispielhaft verdeutlichen. Wir erheben deshalb hier keinen Anspruch auf vollständige Stoffüberdeckung. In der realen Klausur können und werden also auch andere Stoffschwerpunkte geprüft werden. Weiterhin kann die Zusammensetzung ('Quickies', Textaufgaben, Handsimulationen, Programmieraufgaben) variieren.

Diese Probeklausur soll Ihnen helfen, sich mit der Herangehensweise zum Lösen der Aufgaben vertraut zu machen. Es nutzt Ihnen nichts, wenn Sie die Aufgaben oder Lösungen dieser Probeklausur auswendig lernen. Prüfungsrelevant ist der in der Veranstaltung (VL und Übungen) gelehrt Stoff!

Punkte	
--------	--

Name:

Matrikelnummer:

Aufgabe 1: Bäume und Graphen

(4 Punkte)

(a) (1 Punkt) Definieren Sie den Begriff *stark zusammenhängend* für gerichtete Graphen.

(b) (1 Punkt) Wieviele Kanten hat ein Binärbaum mit n Knoten? Geben Sie die Anzahl genau an.

(c) (1 Punkt) Wieviele Kanten hat ein ungerichteter Graph ohne Schlingen höchstens (Eine Schlinge ist eine Kante, die einen Knoten mit sich selbst verbindet.)? Geben Sie die Anzahl möglichst genau an (kein \mathcal{O} -Kalkül!).

(d) (1 Punkt) Definieren Sie den Begriff *AVL-Baum*.

Punkte

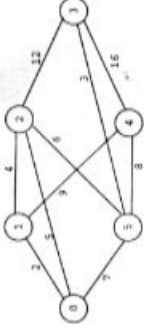
Name:

Matrikelnummer:

Aufgabe 2: Graphen - Minimaler Spannbaum

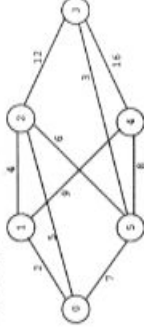
(5 Punkte)

Führen Sie von Hand auf dem abgebildeten Graphen Prim's Algorithmus zur Berechnung eines minimalen Spannbaumes aus.

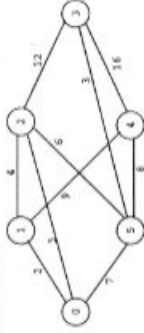


Markieren Sie in den folgenden Graphen die Spannbaum-Kanten in der Reihenfolge, wie sie von Prim's Algorithmus gewählt werden. Startknoten ist Knoten 0.

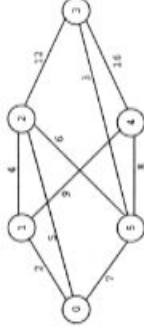
1. Kante



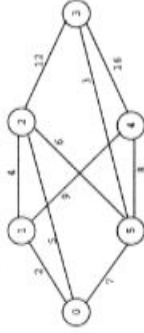
2. Kante



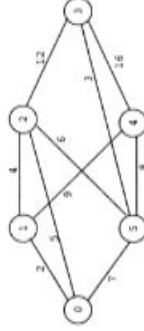
3. Kante



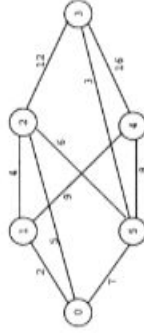
4. Kante



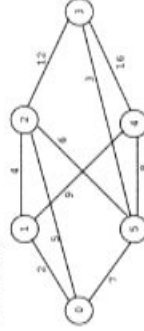
5. Kante



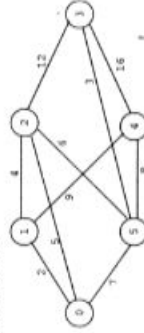
6. Kante



7. Kante



8. Kante



Punkte

Name: _____

Matrikelnummer: _____

Name: _____

Matrikelnummer: _____

Aufgabe 3: Hashtabellen (13 Punkte)

Ein Kleinbetrieb möchte sein Artikelsortiment neuerdings mit dem Rechner verwalten. Dazu wurde jedes Produkt mit einer eindeutigen, natürlichen Zahl als Artikelnummer versehen.

- (a) (6 Punkte) Die Artikel sollen zunächst in einer Hashtabelle T mit der Größe $n = 13$ mit der Hashfunktion $h(k) = km \bmod n$ verwaltet werden. Eventuelle Kollisionen sind durch *lineare Fortschaltung* mit dem Inkrement $a = 3$ aufzulösen.

Füllen Sie die anfangs leere Hashtabelle T , indem Sie der Reihe nach die Artikelnummern 30, 50, 40, 17, 59, 70, 45, 55, 76, 10, 47, 38 eintragen. Benutzen Sie dafür die unten vorbereitete Tabelle. Geben Sie außerdem für jede Artikelnummer die Folge der beim Einfügen durchlaufenen Tabellenplätze an.

Index	Artikelnummer	Folge der durchlaufenen Tabellenplätze
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		

- (b) (7 Punkte) Schreiben Sie in Java eine Methode, die für das Hashverfahren aus Aufgabenteil a) eine Löschfunktionalität implementiert. Die Übergabeparameter für die Methode sollen die Hashfunktion und die Artikelnummer des zu löschenden Artikels sein, als Resultatwert soll die resultierende Hashtabelle zurückgegeben werden.

HINWEIS: Entwerfen Sie zunächst eine geeignete Datenstruktur für die zu speichernden Elemente (Sie brauchen keine Konstruktoren etc. angeben!).

HINWEIS: Verschieben Sie keine Elemente, sondern benutzen Sie ein zusätzliches eindeutiges Element als Platzhalter für die gelöschten Einträge.

	Punkte
--	--------

	Punkte
--	--------