

## Hinweise zur Bearbeitung der Prüfungsklausur zum Modul 63113 „Datenstrukturen und Algorithmen“

Bitte **lesen Sie** sich diese Hinweise **vollständig und aufmerksam** durch, bevor Sie mit der Bearbeitung der Klausur beginnen.

1. Die **Klausurdauer** beträgt **2 Stunden**.
2. Es handelt sich um eine **Auswahlklausur**. Das bedeutet, dass Sie nur einen Teil der Aufgaben lösen müssen. Theoretisch sind **maximal 150 Punkte** erreichbar. Ab **100 Punkten** erhalten Sie die **Bestnote**. Zum **Bestehen** der Klausur sind **50 Punkte** erforderlich.
3. Für die Klausur sind **keine Hilfsmittel zugelassen**. Es darf nur unbeschriebenes Konzeptpapier und Schreibzeug verwendet werden. Die Reinschrift der Klausur darf **nicht mit Bleistift** erfolgen.
4. Schreiben Sie Ihre Lösungen auf Ihr **eigenes Papier** (DIN A4) und nicht auf die Seiten mit den Aufgabenstellungen. Dabei muss **jede Seite** Ihrer Lösung mit **Ihrem Namen** (oben links) und **Ihrer Matrikelnummer** (oben rechts) beschriftet sein. **Nummerieren Sie** bitte alle Seiten Ihrer Lösung.
5. Bevor Sie mit der **Bearbeitung der Klausuraufgaben** beginnen, füllen Sie bitte die folgenden Teile der Klausur aus (sofern das noch nicht geschehen ist):
  - das **Deckblatt mit Name sowie Matrikelnummer**
  - die **Selbstständigkeitserklärung** (beide Dokumente sind seit ca. zwei Wochen abrufbar als Aufgabe „Deckblatt & Erklärung“)
6. **Wenden Sie** bei der Lösung der Aufgaben, soweit dies möglich ist, die **Algorithmen und Notationen aus den Kurseinheiten an**.
7. **Vermeiden Sie** in jedem Fall bei der **Definition von Funktionen in Algebren Java-Programme sowie Algorithmen**. Geben Sie lediglich mathematische Definitionen an, wie sie im Kurstext verwandt worden sind!  
Ebenso sind **Mengendefinitionen in mathematischer Mengennotation** durchzuführen. Verwenden Sie **weder programmiersprachliche Definitionen noch konkrete Datenstrukturen!**
8. **Abgabe Ihrer Klausur:**
  - **Scannen** Sie Ihre Lösungen ein und erstellen Sie genau **ein PDF-Dokument**, das alle Seiten umfasst.
  - **Laden** Sie das Lösungsdokument **bis spätestens 19:30 Uhr** im Online-Übungssystem hoch (dort, wo Sie die Klausuraufgaben heruntergeladen haben).
  - Sollte der Upload **fehlschlagen** (und nur dann), senden Sie Ihre Lösung per **E-Mail** an [fabio.valdes@fernuni-hagen.de](mailto:fabio.valdes@fernuni-hagen.de).
  - **Sichern** Sie Ihre eingescannte Lösung zusätzlich auf einer Festplatte.
  - **Kreuzen** Sie auf dem Deckblatt die von Ihnen **bearbeiteten Aufgaben an**.
  - **Senden** Sie Ihre **Lösungsblätter mit Deckblatt und Selbstständigkeitserklärung** im Original per Einschreiben an die folgende Anschrift:  

FernUniversität in Hagen  
Fakultät für Mathematik und Informatik  
Prüfungsamt (IZ)  
58084 Hagen
  - **Beschriften** Sie den Umschlag zusätzlich mit „Klausur 63113“ oder „Klausur Datenstrukturen“.

## Aufgabe 1 Produkt-Algebra

24 Punkte

In dieser Aufgabe soll ein Ausschnitt eines Produktinformationssystems mit Hilfe einer Algebra modelliert werden. Ein Produktinformationssystem verwaltet verschiedene Produkte sowie Produktdaten wie marketingrelevante oder technische Informationen. Diese Daten können z. B. von Händlern für ihre E-Commerce-Aktivitäten genutzt werden.

Produkte werden in Katalogen organisiert, d. h., ein Katalog enthält eine Menge von Produkten. In dieser Algebra wird ein Produkt durch eine eindeutige  $ID$ , geplante Verkäufe (Menge von Tupeln) und den Lagerbestand (ganzzahlig) beschrieben. Die Menge der geplanten Verkäufe enthält zweielementige Tupel (Paare), wobei das erste Element des Tupels einen Ländercode und das zweite die geplante Verkaufszahl darstellt (beide ganzzahlig). Unternehmen erstellen geplante Verkaufszahlen, um Verkäufe länderspezifisch zu planen und diese mit regionalen Händlern abzustimmen. Ändert sich die Marktlage, dann werden die geplante Verkaufszahlen angepasst.

Die Sorten und Trägermengen der Algebra sind wie folgt vorgegeben (der Identifikator  $ID$  wird nicht näher spezifiziert und direkt verwendet):

<b>algebra</b>	<i>produktverwaltung</i>	
<b>sorts</b>	$int_0, int_+, ID, geplanteVerkäufe, produkt, katalog$	
<b>sets</b>	$int_0$	$= \{n \in int \mid n \geq 0\}$
	$int_+$	$= \{n \in int \mid n > 0\}$
	$geplanteVerkäufe$	$= \{V \subseteq int_+ \times int_+ \mid V \text{ endlich}\}$
	$produkt$	$= ID \times geplanteVerkäufe \times int_0$
	$katalog$	$= \{K \subseteq produkt \mid K \text{ endlich}\}$

Die Algebra soll die folgenden Operationen unterstützen:

**neuerKatalog** Erzeugt einen leeren Katalog.

**neuesProdukt** Erzeugt ein neues Produkt ohne geplante Verkäufe.

**produktEinfügen** Fügt ein Produkt dem Katalog hinzu, falls die  $ID$  des einzufügenden Produktes mit keiner Produkt- $ID$  eines im Katalog vorhandenen Produktes übereinstimmt.

**verkäufeÄndern** Zu einem gegebenen Produkt und einem gegebenen Verkaufszahlen-Tupel gilt Folgendes: Falls in der Menge der Verkaufszahlen des Produktes kein Tupel existiert, dessen Ländercode mit dem Ländercode aus dem gegebenen Tupel übereinstimmt, dann wird das gegebene Tupel zu der Menge der Verkaufszahlen hinzugefügt. Ansonsten wird das vorhandene Tupel ersetzt.

**bestandÜberErmitteln** Diese Operation liefert eine Menge von Produkten aus einem Katalog, deren Lagerbestand über einem bestimmten Wert liegt.

**gesamtbestandBerechnen** Diese Operation berechnet die Gesamtzahl aller im Lager befindlichen Produkte eines Kataloges.

**produktOhneBestandLöschen** Entfernt ein Produkt aus dem Katalog, falls sein Bestand den Wert 0 hat.

**katalogLeeren** Löscht alle Produkte aus einem Katalog.

- (a) Geben Sie die Signaturen für die oben beschriebenen Operationen an. Für jede korrekte Signatur gibt es einen Punkt. 8 Punkte
- (b) Geben Sie Funktionsdefinitionen für die Operationen der Algebra an. Pro Definition sind maximal zwei Punkte erreichbar. 16 Punkte

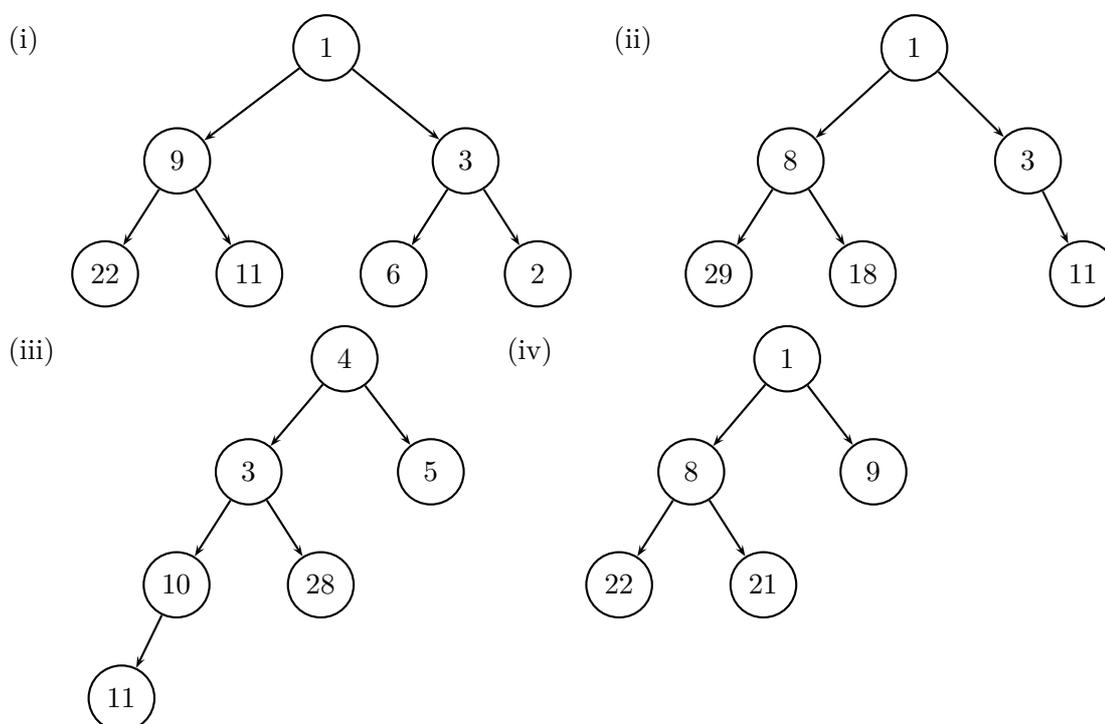
HINWEISE:

- Führen Sie bei Bedarf Hilfsmengen oder Hilfsoperatoren ein, um die Aufgabe strukturiert lösen zu können. Diese müssen formal (also mit Signatur und Funktion) angegeben werden.
- Sie können die Operation  $\pi_i(X)$  verwenden, um aus einem Tupel  $X$  die  $i$ -te Komponente zu erhalten.
- Die einfachen Sorten *bool*, *int*, *real* sowie *string* können Sie verwenden, ohne sie definieren zu müssen.

## Aufgabe 2 Heaps

24 Punkte

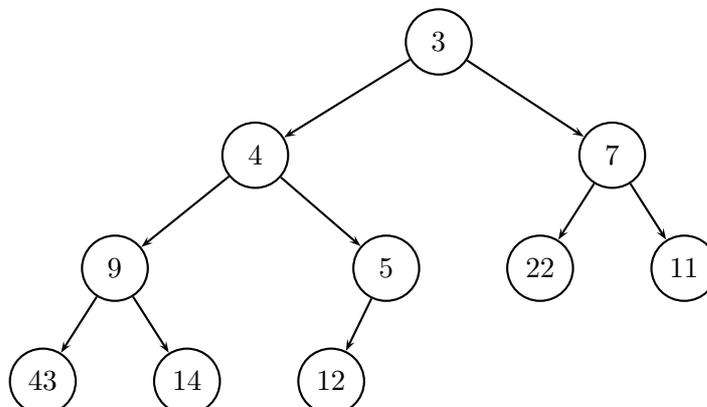
- (a) Gegeben sind folgende Binärbäume. Welche der Abbildungen stellen einen Minimum-Heap bzw. einen links-vollständigen Minimum-Heap dar? Welche Bäume sind keine Heaps? Begründen Sie jeweils Ihre Antworten. 4 Punkte



- (b) Fügen Sie die Zahlen 10 Punkte  
 4, 11, 1, 42, 17, 7, 6, 3, 2, 5

in der angegebenen Reihenfolge in einen anfangs leeren Minimum-Heap ein. Zeichnen Sie den Baum nach jedem Einfügevorgang. Alternativ dazu können Sie auch die Array-Darstellung verwenden.

- (c) Entnehmen Sie aus dem unten dargestellten Minimum-Heap fünfmal hintereinander das minimale Element. Stellen Sie den Heap nach jedem Löschvorgang dar. Zeichnen Sie dazu den entsprechenden Baum, oder verwenden Sie die Darstellung im Array. 10 Punkte



### Aufgabe 3 Sortieren

24 Punkte

- (a) Gegeben sei die Zahlenfolge:

12 Punkte

42, 21, 25, 33, 13, 12, 3, 8, 30, 22

Sortieren Sie diese Folge *aufsteigend* mit Hilfe von Quicksort. Geben Sie dazu den Baum der rekursiven Aufrufe an. Jeder Knoten des Baum entspricht dabei einem Aufruf von Quicksort und enthält die Eingabefolge des Aufrufs. Die Kinder eines Knotens entsprechen den beiden rekursiven Aufrufen. Der Merge-Schritt wird nicht dargestellt. Markieren Sie in jedem Knoten das Pivotelement, das mit der Funktion *findx* aus dem Kurstext zu bestimmen ist.

- (b) Gegeben sei die Zahlenfolge:

12 Punkte

6, 16, 1, 9, 38, 18, 8, 36, 42, 11

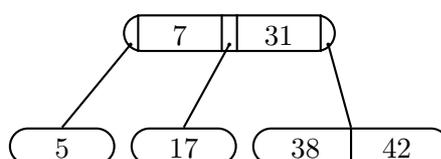
Sortieren Sie diese Folge *aufsteigend* mittels Standard-Heapsort. Geben Sie den initialen Heap an. Stellen Sie weiterhin den Heap und die sortierte Folge nach jedem Einsinkenlassen eines Elements dar. Sie können den Heap wahlweise im Array oder als Baum darstellen.

### Aufgabe 4 B-Baum

20 Punkte

- (a) Gegeben sei der folgende B-Baum der Ordnung 1.

10 Punkte

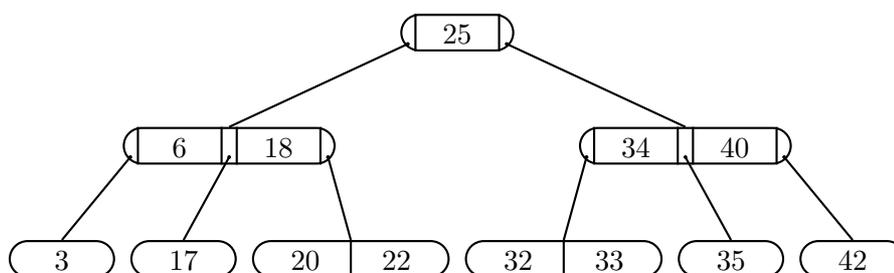


Fügen Sie in obigen B-Baum die folgenden Schlüsselwerte in der gegebenen Reihenfolge ein. Geben Sie den Baum jeweils vor und nach einer Split-Operation an.

28 29 25 27 34

(b) Gegeben sei der folgende B-Baum der Ordnung 1:

10 Punkte

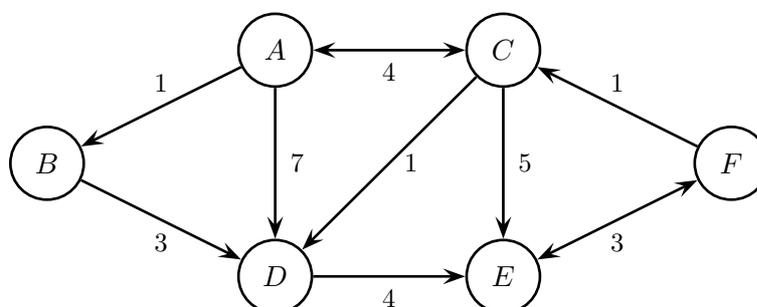


Löschen Sie aus diesem B-Baum nacheinander die Elemente 6, 3, 22 und 17 (in dieser Reihenfolge). Geben Sie bei einem Unterlauf jeweils den Baum vor und nach der Unterlaufbehandlung an und benennen Sie die Art der Unterlaufbehandlung.

## Aufgabe 5 Dijkstra

18 Punkte

Gegeben sei der folgende Graph :

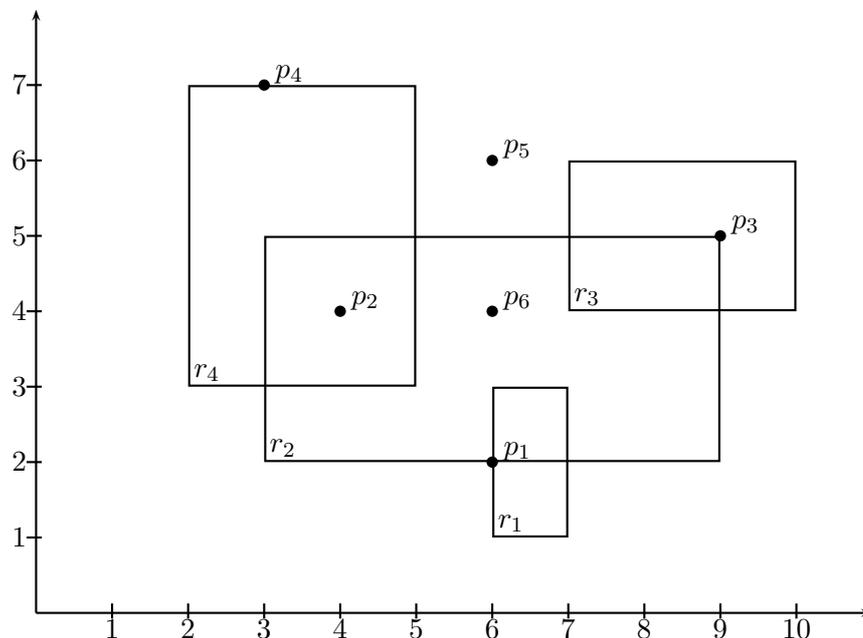


Bestimmen Sie mit Hilfe des Algorithmus von Dijkstra die kürzesten Wege vom Knoten  $A$  zu allen anderen Knoten. Sie können die Aufgabe wahlweise graphisch, textuell oder tabellarisch lösen. Für jeden Schritt des Algorithmus muss erkennbar sein

- welcher Knoten grün gefärbt wird,
- welche Knoten gelb gefärbt werden und wie lang der kürzeste Weg dorthin ist,
- welche Kanten rot gefärbt werden,
- welche Kanten gelb gefärbt werden,
- zu welchen Knoten neue Kosten ermittelt wurden und wie hoch diese sind.

**Aufgabe 6 Punkteinschluss****20 Punkte**

Gegeben seien die orthogonalen Rechtecke  $r_1$  bis  $r_4$  sowie die Punkte  $p_1$  bis  $p_6$  in der folgenden Konfiguration:



Lösen Sie das Punkteinschlussproblem für diese Mengen mittels des Plane-Sweep-Algorithmus aus dem Kurstext. Führen Sie dazu die folgenden Schritte durch:

- Geben Sie den leeren Segmentbaum an, der für diese Mengen benötigt wird. *6 Punkte*
- Geben Sie alle auftretenden Events entsprechend der Reihenfolge der Verarbeitung an. Achten Sie darauf, dass auch Punkte auf Rändern von Rechtecken gefunden werden sollen. *7 Punkte*
- Führen Sie den Algorithmus schrittweise aus und erklären Sie kurz, was in den einzelnen Schritten passiert. *7 Punkte*

**Aufgabe 7 Externes Sortieren****20 Punkte**

Gegeben sei die Eingabefolge: 21, 3, 17, 29, 38, 5, 2, 71, 16, 25, 55, 99, 12.

- Sortieren Sie die Folge durch direktes Mischen ohne internes Sortieren. Erzeugen Sie zunächst Läufe der Länge 1. *6 Punkte*
- Sortieren Sie die Folge durch direktes Mischen mit internem Sortieren. Wählen Sie dabei  $m = 3$ . *6 Punkte*
- Sortieren Sie die Folge durch natürliches Mischen mit Ersetzungs-Auswahl, wobei der Hauptspeicher 3 Sätze speichern kann. *8 Punkte*