

Hinweise zur Bearbeitung der Klausur zum Kurs 1661 Datenstrukturen I

Bitte lesen Sie sich diese Hinweise vollständig und aufmerksam durch, bevor Sie mit der Bearbeitung der Klausur beginnen. Beachten Sie insbesondere die Punkte 7 und 8!

1. Die Klausurdauer beträgt 2 Stunden.
2. Prüfen Sie bitte die Vollständigkeit Ihrer Unterlagen. Die Klausur umfasst:
 - 2 Deckblätter
 - diese Hinweise
 - 1 Formblatt für eine Teilnahmebescheinigung zur Vorlage beim Finanzamt
 - 5 Aufgaben auf den Seiten 2–4
3. Bevor Sie mit der Bearbeitung der Klausuraufgaben beginnen, füllen Sie bitte die folgenden Teile der Klausur aus:
 - (a) sämtliche Deckblätter mit Name, Anschrift sowie Matrikelnummer. Markieren Sie vor der Abgabe auf allen Deckblättern die von Ihnen bearbeiteten Aufgaben.
 - (b) die Teilnahmebescheinigung, falls Sie diese wünschen.
4. Schreiben Sie Ihre Lösungen auf Ihr eigenes Papier (DIN A4) und nicht auf die Seiten mit den Aufgabenstellungen. Heften Sie vor Abgabe der Klausur die Deckblätter (und evtl. die Teilnahmebescheinigung) an Ihre Bearbeitung.
5. Schreiben Sie bitte auf jedes Blatt oben links Ihren Namen und oben rechts Ihre Matrikelnummer. Nummerieren Sie Ihre Seiten bitte durch.
6. Wenden Sie bei der Lösung der Aufgaben, soweit dies möglich ist, die Algorithmen und Notationen aus den Kurseinheiten an.
7. Schreiben Sie, wenn Algorithmen gefordert sind, keine kompletten Java-Programme, sondern beschränken Sie sich auf die wesentlichen Teile des Algorithmus, die z.T. auch in Prosa formuliert werden können. Formulieren Sie Algorithmen aber so, dass die elementaren Einzelschritte erkennbar werden.

Sparen Sie bei solchen Aufgaben nicht mit Kommentaren. Wenn Ihre Lösung aufgrund fehlender Kommentare nicht verständlich ist, führt das zu Punktabzug.
8. 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. Geben Sie auf keinen Fall Datentypdefinitionen an, und verwenden Sie keine konkreten Datenstrukturen!
9. Als Hilfsmittel sind nur unbeschriebenes Konzeptpapier und Schreibzeug zugelassen. Die Reinschrift der Klausur darf **nicht mit Bleistift** erfolgen.
10. Durch Lösen der Aufgaben sind maximal 100 Punkte erreichbar. Sie dürfen damit rechnen einen Übungsschein bzw. ein Zertifikat zu erhalten, wenn Sie insgesamt mindestens 50 Punkte erreichen.

25 Punkte

Aufgabe 1 Buchalgebra

In dieser Aufgabe sollen Sie eine Algebra zur Verwaltung einer Literatursammlung erstellen. Eine Literatursammlung besteht aus Büchern. Jedes Buch enthält mindestens eine Geschichte. Ein Buch kann aber auch mehrere verschiedene Geschichten enthalten. Enthält das Buch mehrere Geschichten, dürfen diese nicht gleich sein. Jede Geschichte hat einen Titel, genau einen Autor und eine Seitenanzahl. Jedes Buch hat eine eindeutige Identifikationsnummer (ISBN), einen Titel und eine Auflistung der im Buch enthaltenen Geschichten. Zur Vereinfachung dürfen Sie davon ausgehen, dass ISBN-Nummern einfach positive ganze Zahlen sind. Die Sammlung darf jedes Buch nur einmal enthalten.

Die Algebra umfasst neben den notwendigen Sorten auch folgende Operationen:

- / *createStory*: Erzeugt aus einem Titel, einem Autor und einer Seitenzahl eine Geschichte.
- / *createBook*: Erzeugt aus einer ISBN, einem Titel und einer Geschichte ein Buch.
- / *createSet*: Erzeugt eine leere Literatursammlung.
- insertStory*: Fügt eine Geschichte in ein Buch ein.
- insertBook*: Fügt ein Buch in eine Sammlung ein.
- deleteBook*: Entfernt ein Buch aus einer Sammlung.
- deleteStory*: Entfernt eine Geschichte aus einem Buch, wenn sie darin enthalten ist und nicht die einzige Geschichte im Buch ist.
- / *equalBooks*: Prüft, ob zwei Bücher gleich sind.
- / *equalStories*: Prüft, ob zwei Geschichten gleich sind.
- / *containsStory*: Prüft, ob eine Geschichte in einem Buch vorhanden ist.
- / *containsBook*: Prüft, ob ein Buch in einer Sammlung vorhanden ist.
- *findStory*: Gibt, wenn die Geschichte in einer Literatursammlung vorhanden ist, alle Bücher, die die Geschichte enthalten, aus.

Formulieren Sie die Spezifikation aller erforderlichen Datentypen und Operationen als Algebra.

4 Punkte

(a) Geben Sie die Sorten und Operatorsignaturen der Algebra an.

Sort
ops

6 Punkte

(b) Geben Sie Trägermengen für die Sorten an, die genau die geforderten Objekte enthalten. Dabei können Sie *bool* und *string* als gegeben voraussetzen.

func

15 Punkte

(c) Geben Sie die Funktionen für die Operationen mit der oben aufgeführten Semantik an.

sets

Aufgabe 2 AVL-Baum**26 Punkte**

- /(a) Fügen Sie in einen zunächst leeren AVL-Baum die nachstehenden Elemente in der gegebenen Reihenfolge ein. Zeichnen Sie den Baum vor und nach jeder Rebalancierung. Markieren Sie jeweils den Knoten, der aus der Balance geraten ist und geben Sie die jeweilige Balancierungsmethode an. 7 Punkte

11, 22, 15, 24, 7, 2, 23

11 15 22 24
7

- /(b) Beschreiben Sie die generelle Vorgehensweise zur Konstruktion eines minimal-gefüllten AVL-Baumes der Höhe h . 3 Punkte
- (c) Geben Sie einen laufzeit-optimalen Algorithmus *arrayToAVLtree* an, der aus einem aufsteigend sortierten Array A mit n verschiedenen Schlüsseln einen AVL-Baum erzeugt. 8 Punkte
- (d) Beweisen Sie die Korrektheit Ihres Algorithmus' aus Teilaufgabe (c). 8 Punkte

Aufgabe 3 Sortierverfahren**23 Punkte**

Paul soll eine Reihe nebeneinander aufgestellter Kisten nach Größe aufsteigend sortieren. Er beginnt am linken Ende der Reihe und vergleicht die Kiste, vor der er steht, mit ihrem rechten Nachbarn. Ist der rechte Nachbar kleiner, vertauscht Paul die Kisten miteinander und geht eine Kiste weiter nach links. Sollte dies nicht möglich sein, weil Paul schon am linken Ende der Reihe steht, geht er einen Schritt nach rechts. Ist beim Vergleichen der rechte Nachbar nicht kleiner, geht Paul eine Kiste weiter nach rechts. Wenn Paul am rechten Ende der Reihe angekommen ist, hört er auf zu arbeiten und geht nach Hause.

- /(a) Sind die Kisten korrekt sortiert, wenn Paul nach Hause geht? Begründen Sie Ihre Antwort. 3 Punkte
- /(b) Welche Datenstruktur können Sie verwenden, um das oben beschriebene Verfahren am PC umzusetzen? 1 Punkt
- /(c) Geben Sie einen formalen Algorithmus für das oben beschriebene Sortierverfahren an, der die unter (b) genannte Datenstruktur nutzt. 5 Punkte
- /(d) Welche Laufzeit (O-Notation) hat das oben beschriebene Verfahren im besten Fall? Und welche Laufzeit (O-Notation) hat es im schlimmsten Fall? 2 Punkte
- /(e) Ist das Verfahren optimal? Begründen Sie Ihre Antwort. 2 Punkte
- (f) Klassifizieren Sie das oben angegebene Sortierverfahren nach den Kriterien des Kurstextes. 4 Punkte

/ 6 Punkte

- (g) Sortieren Sie die unten angegebene Zahlenfolge *absteigend* mittels Mergesort. Es reicht aus, wenn Sie als Lösung den Aufrufbaum des Mergeschrittes angeben.

15 - 96 - 47 - 32 - 43 | 18 - 5 - 68 - 103 | 55 - 27 - 38 - 34 - 1 - 82 - 135

25 Punkte

Aufgabe 4 Dijkstra-Modifikationen

15 Punkte

- (a) Im Kurs haben Sie den Algorithmus von Dijkstra zur Bestimmung kürzester Wege von einem Knoten in einem Graphen G zu allen anderen Knoten kennengelernt. In dieser Aufgabe soll bestimmt werden, ob ein Knoten t in einem gegebenen Graphen G von einem Knoten s des Graphen mit höchstens Kosten k erreicht werden kann. Natürlich könnte man mittels des Dijkstra-Algorithmus' die Kosten von s zu allen Knoten bestimmen lassen und dann prüfen, ob die Kosten geringer sind als vorgegeben. Dies würde jedoch viele unnötige Berechnungen bedeuten.

Wie kann man den Algorithmus von Dijkstra ändern, so dass die Berechnung abgebrochen wird, sobald bekannt ist, ob der Knoten mit höchstens den gegebenen Kosten erreicht oder nicht erreicht werden kann.

Geben Sie die modifizierte Version des Algorithmus' an.

10 Punkte

- (b) Ein anderes Problem ist das Finden aller Knoten eines Graphen, die mit höchstens Kosten k von einem gegebenen Knoten aus erreicht werden können ohne den gesamten Graphen zu durchsuchen (ausser, alle Knoten sind mit Kosten k erreichbar). Geben Sie die dazu notwendigen Änderungen des Dijkstra-Algorithmus' an (der Algorithmus selbst muss in dieser Teilaufgabe nicht angegeben werden).

1 Punkt

Aufgabe 5 Deckblatt

Lesen Sie sich die „Hinweise zur Bearbeitung“ sorgfältig durch. Füllen Sie den dort aufgeführten Anweisungen entsprechend beide Deckblätter vollständig und korrekt aus.