

**Hinweise zur Bearbeitung der Klausur zum Kurs 1662/1663 Datenstrukturen**

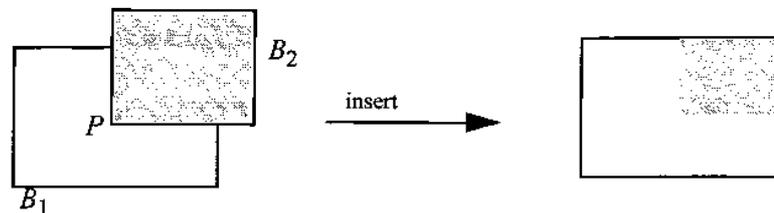
Bitte lesen Sie sich diese Hinweise vollständig und aufmerksam durch, bevor Sie mit der Bearbeitung der Klausur beginnen. Beachten Sie insbesondere die Punkte 7 und 8!

1. Die Klausurdauer beträgt 3 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
  - 6 Aufgaben auf den Seiten 2 – 5
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. Numerieren 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 PASCAL- oder 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 aber 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* PASCAL- oder 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 (kein Bleistift!) zugelassen.
10. Es sind maximal 100 Punkte erreichbar. Zur Erlangung eines Übungsscheins bzw. Zertifikats benötigen Sie etwa 50 Punkte.

## 22 Punkte Aufgabe 1 Algebra für Bilder

In dieser Aufgabe soll eine Algebra für Bilder entworfen werden. Jedes Bild hat eine Höhe  $h$  sowie eine Breite  $b$  (positive ganze Zahlen). Jeder Bildposition  $\{(x, y) \mid 1 \leq x \leq b, 1 \leq y \leq h\}$  sei eine Farbe  $c_{xy}$  zugeordnet. Die Position  $(1, 1)$  bezeichne dabei die linke untere Ecke des Bildes. Jede Farbe sei durch ihr Mischverhältnis von Rot, Grün und Blau im Bereich von 0 bis 1 beschrieben. Sind alle Teile 0, so entsteht die Farbe Schwarz. Weiß erhält man, wenn alle Komponenten auf 1 gesetzt werden. Die Algebra soll folgende Operationen unterstützen:

- *createPicture*: erzeugt ein weißes Bild mit gegebener Größe
- *insert*: fügt ein Bild  $B_2$  in ein anderes Bild  $B_1$  an einer gegebenen Position ein. Dabei werden die überdeckten Bildpunkte des alten Bildes überschrieben. Es werden nur Positionen innerhalb von  $B_1$  berücksichtigt. Die Positionsangabe beschreibt die linke untere Koordinate, an der das neue Bild beginnen soll:



- *rotate*: dreht ein Bild um 90 Grad gegen den Uhrzeigersinn
- *height*: gibt die Höhe eines Bildes zurück
- *width*: gibt die Breite eines Bildes zurück
- *getColor*: gibt die Farbe an einer gegebenen Position zurück
- *setColor*: ändert die Farbe der gegebenen Position
- *mirror*: spiegelt das Bild an der vertikalen Achse in der Mitte des Bildes
- *cut*: erzeugt den Ausschnitt des Bildes, welches durch das achsenparallele Rechteck, das durch zwei Positionen gegeben ist, begrenzt wird.

*Hinweise:*

Die Funktionen *getColor* und *cut* produzieren jeweils einen undefinierten Wert, wenn die Position bzw. der Bereich außerhalb des Bildes liegt, welches als Argument übergeben wird. Überlegen Sie sich, welche Auswirkungen dies auf die Trägermengen sowie alle anderen Operatoren hat.

Ein zweidimensionales Array der Größe  $n \times m$  läßt sich in mathematischer Notation durch eine Folge von  $n \cdot m$  Elementen darstellen. Diese können beispielsweise durch ihre Position im Array  $(i, j)$  indiziert werden.

- Identifizieren Sie die Sorten der Algebra und geben Sie die Signaturen aller Operationen an.
- Definieren Sie die Trägermengen sowie die Funktionen der Algebra.
- Geben Sie einige Axiome (mind. 5) für die Algebra an, so daß die Operationen *createPicture*, *width*, *height*, *insert*, *getColor*, *setColor*, *rotate* und *mirror* mindestens einmal verwendet werden.

**Aufgabe 2 Hashing****12 Punkte**

Fügen Sie die aufgeführte Zahlenfolge in eine Hashtabelle mit  $m=10$  unter Zuhilfenahme quadratischen Sondierens (Grundform) und der Mittel-Quadrat-Methode als Hashfunktion ein. Geben Sie für jede eingefügte Zahl die Folge der Behälternummern an, die getestet werden. Wie lautet der Inhalt der finalen Hashtabelle?

120 - 163 - 19 - 84 - 69 - 67 - 175 - 59 - 13

**Aufgabe 3 Radixsort****14 Punkte**

- (a) Es sei *keytype* die Menge aller Zeitstempel der Form *hh:mm:ss* mit  $hh \in \{00, 01, \dots, 23\}$  und  $mm, ss \in \{00, 01, \dots, 59\}$ . Sortieren Sie die unten angegebene Folge mittels Radixsort in drei Phasen.

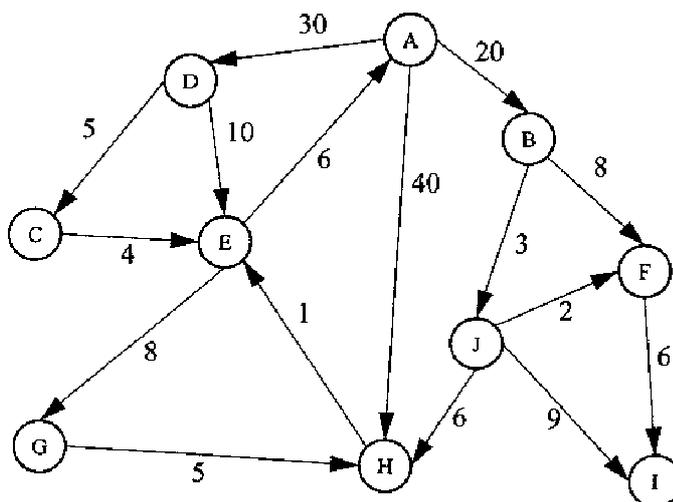
**8 Punkte**

23:59:01, 13:31:14, 12:11:01, 03:27:01, 17:23:14, 12:11:03, 17:15:59, 13:28:01

- (b) Wenn Sie 1000 Zahlen aus dem Bereich  $1 - 10^{100}$  sortieren sollen, würden Sie Heapsort oder Radixsort bevorzugen? Begründen Sie Ihre Antwort.

**6 Punkte****Aufgabe 4 Bestimmung kürzester Wege in Graphen****20 Punkte**

Bestimmen Sie die kürzesten Wege vom Knoten *A* zu allen anderen Knoten im unten angegebenen Graphen unter der Verwendung des Algorithmus von Dijkstra. Markieren Sie in jedem Schritt die Menge der gelben und grünen Knoten sowie die roten und gelben Kanten geeignet. Gibt es mehrere Knoten zur Auswahl, wählen Sie den alphabetisch kleineren. Ergeben sich durch das Grünfärben eines Knotens keine Änderungen bei den Kanten, so muß der Graph nicht erneut gezeichnet werden.



**18 Punkte Aufgabe 5 Geometrische Algorithmen**

2 Punkte (a) Formulieren Sie das Punkteinschluß-Problem. Wie viele Punkteinschlüsse können maximal vorkommen?

12 Punkte (b) Geben Sie den im Kurstext vorgestellten Algorithmus zur Lösung des Punkteinschluß-Problems an. Im Algorithmus sollen die Datenstrukturen *SE*, *SS* und *B* mit folgenden Funktionen benutzt werden:

<i>SE:</i>	<i>append</i>	Fügt ein Element in die Sweep-Event-Struktur ein.
	<i>sort</i>	Sortiert die enthaltenen Elemente.
	<i>startScan</i>	Initialisiert einen sequentiellen Durchlauf durch die Elementliste.
	<i>getNext</i>	Liefert das nächste Element oder den Wert <i>null</i> .
<i>SS:</i>	<i>insert</i>	Fügt ein Element in die Sweep-Status-Struktur ein.
	<i>remove</i>	Entfernt ein Element aus der Sweep-Status-Struktur.
	<i>find</i>	Startet eine Suche auf der Status-Struktur.
	<i>getNext</i>	Liefert das nächste gefundene Element oder den Wert <i>null</i> .
<i>B:</i>	<i>insert</i>	Fügt ein Paar (Schlüsselwert, Rechteck) in einen B-Baum ein.
	<i>get</i>	Gibt das unter einem Schlüsselwert gespeicherte Rechteck zurück.

Diese Datenstrukturen unterstützen eine persistente Speicherung, daher muss z.B. über die Lösungsmenge einer *find* Operation mittels *getNext* iteriert werden. Eingabeparameter für den Algorithmus sind eine Punktmenge *P* und eine Rechteckmenge *R*.

4 Punkte (c) Wie würden Sie vorgehen, um die von einer Menge achsenparalleler Rechtecke mehrfach überdeckte Fläche, das heißt Summe der Flächen, die von zwei oder mehr Rechtecken überdeckt werden, mittels Rechteckschnitt und Maßproblem zu berechnen? Geben Sie die Laufzeit für Ihre Lösung an.

**14 Punkte Aufgabe 6 Externes Sortieren**

7 Punkte (a) Sortieren Sie die unten angegebene Folge mittels Vielweg-Mischen für  $k=3$ :  
43, 71, 12, 35, 11, 80, 99, 47, 62, 15, 77, 26, 69, 93, 19, 45, 23, 38, 56, 91, 10  
Berechnen Sie vorher die Anzahl der benötigten Mischphasen!

- (b) Im Kurstext wurden meistens Zahlenfolgen sortiert. Die Verfahren funktionieren allerdings für alle Datentypen, die eine Ordnung besitzen. Angenommen ein Vergleich benötigt genauso viel Zeit wie das Lesen und Schreiben der Daten seiner Argumente. Stellen Sie eine Ungleichung auf, die bestimmt, für welches  $k$  das Vielweg-Mischen günstiger ist als direktes Mischen mit Anfangsläufen der Länge 1. Dabei soll jeweils vom *worst case* ausgegangen werden. 7 Punkte

Hinweis: Die Ungleichung muß nicht gelöst werden.