

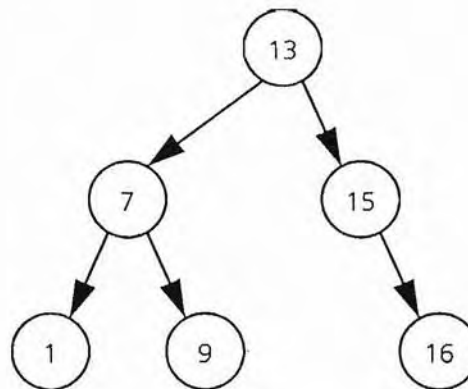
16 Punkte Aufgabe 1 Spezifikationen

Sei K eine Menge, $\oplus: K \times K \rightarrow K$ und $\otimes: K \times K \rightarrow K$ seien zwei Funktionen auf K . $(K, \oplus, \otimes, n, e)$ heißt *Körper* über K , wenn gilt:

1. \oplus ist assoziativ und kommutativ,
2. \otimes ist assoziativ und kommutativ,
3. n (genannt „Nullelement“) ist das neutrale Element bezüglich \oplus ,
4. e (genannt „Einselement“) ist das neutrale Element bezüglich \otimes ,
5. zu jedem Element a aus K existiert ein bezüglich \oplus inverses Element in K (das „Negative“ zu a),
6. zu jedem Element a aus K gibt es ein bezüglich \otimes inverses Element in K (den „Kehrwert“ zu a),
7. für alle Elemente a, b, c aus K gilt das Distributivitätsgesetz:
 $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$.

12 Punkte (a) Geben Sie eine Spezifikation für Körper über beliebige, frei bleibende Mengen K als abstrakten Datentyp an.

4 Punkte (b) Warum können Sie hier stattdessen keine algebraische Spezifikation vornehmen?

15 Punkte Aufgabe 2 AVL-Baum

12 Punkte (a) Fügen Sie die unten angegebene Folge in den oben angegebenen AVL-Baum ein. Zeichnen Sie den Baum dabei vor und nach jeder Rebalancierung und markieren Sie den Knoten, an dem die Balance verletzt wird.

5 - 8 - 10 - 2 - 6 - 4 - 14 - 11 - 3

3 Punkte (b) Löschen Sie aus dem in Aufgabenteil (a) erzeugten Baum die folgenden Elemente in der gegebenen Reihenfolge. Zeichnen sie den Baum nach jedem Löschvorgang neu. Markieren Sie die Knoten, an denen die Balance verletzt wird und zeichnen Sie den Baum nach jeder Rebalancierung neu und geben Sie zum Schluss den Ergebnisbaum an.

13 - 7 - 2

Aufgabe 3 Sortieren**20 Punkte**

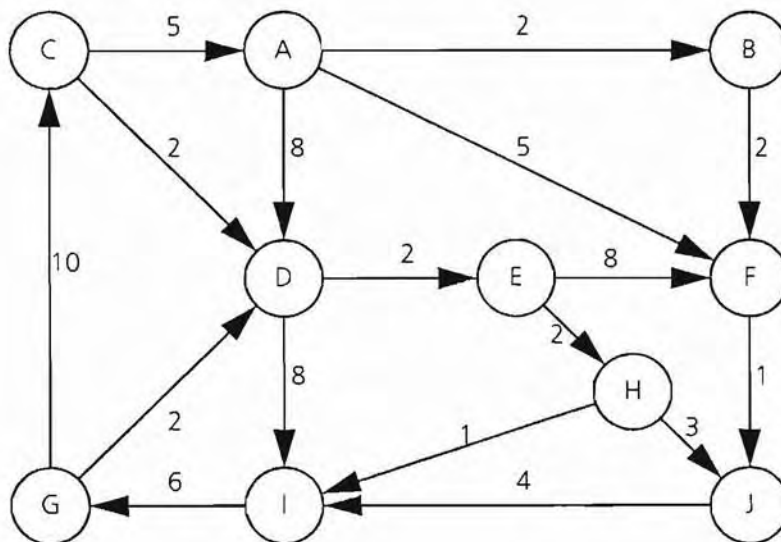
Gegeben sei die Zahlenfolge:

12 - 34 - 54 - 98 - 67 - 55 - 12 - 19 - 37 - 61 - 43 - 77

- (a) Sortieren Sie die Zahlenfolge aufsteigend mittels HeapSort. Zeichnen Sie dazu zunächst die Baumeinbettung dieser Folge, konstruieren Sie dann den initialen Heap und zeichnen diesen wiederum als Baum. Geben Sie nach jedem verarbeiteten Element den Inhalt des Heaps (wahlweise als Baum oder als Array) sowie die bereits sortierte Teilfolge an. 8 Punkte
- (b) Sortieren Sie die Folge mittels MergeSort. Geben Sie dazu den entsprechenden Baum der rekursiven Aufrufe an. 8 Punkte
- (c) Gegeben sei ein Array von n ganzen Zahlen. Geben Sie einen Algorithmus an, der den Inhalt des Arrays so umsortiert, dass alle negativen Zahlen vor den positiven Zahlen stehen. Der Algorithmus soll in situ arbeiten, d.h. kein Hilfsarray verwenden. Geben Sie die Laufzeit für Ihren Algorithmus an (eine formale Begründung ist nicht erforderlich). 4 Punkte

Aufgabe 4 Kürzeste Wege in Graphen**15 Punkte**

Gegeben sei folgender Graph:



Berechnen Sie die kürzesten Wege beginnend bei Knoten A zu allen anderen Knoten. Beschreiben oder zeichnen Sie den Graphen nach jedem Grünfärben eines Knotens. Stehen mehrere Knoten zur Auswahl, soll der lexikografisch kleinere gewählt werden.

Hinweis: Es ist nicht nötig, den Graphen für jeden Schritt des Algorithmus zu zeichnen. Es genügt, wenn Sie beschreiben, was in jedem einzelnen Schritt des Algorithmus passiert. Dazu gehören:

- welcher Knoten wird grün gefärbt
- welche Kanten werden rot gefärbt (in den Baum der minimalen Wege aufgenommen)
- welche Knoten werden gelb gefärbt
- welche Kanten werden gelb gefärbt (gestrichen)
- welche Kostenänderungen gibt es an den Knoten

Sie können natürlich auch den Graphen zeichnen. Markieren Sie in diesem Fall die verschiedenen Knoten- und Kantenklassen geeignet und geben Sie an, welche Markierung zu welcher Klasse gehört.