FernUniversität in Hagen

Lösungsvorschläge zur Hauptklausur „1661 Datenstrukturen I"

9.8.2008

Aufgabe 1

(a)

adt *Körper*

sorts	K			
ops	$\mathbf{e}$	:	$\rightarrow$	K
	$\mathbf{n}$	:	$\rightarrow$	K
	$\oplus$	: $K \times K$	$\rightarrow$	K
	$\otimes$	: $K \times K$	$\rightarrow$	K
	minus	: K	$\rightarrow$	K
	inv	: K	$\rightarrow$	K

axs

$x \oplus y = y \oplus x$

$x \oplus (y \oplus z) = (x \oplus y) \oplus z$

$x \oplus \mathbf{n} = x$

$x \oplus \mathbf{minus}(x) = \mathbf{n}$

$x \otimes y = y \otimes x$

$x \otimes (y \otimes z) = (x \otimes y) \otimes z$

$x \otimes \mathbf{e} = x$

$x \otimes \mathbf{inv}(x) = \mathbf{e}$

$x \otimes (y \oplus z) = (x \otimes y) \oplus (x \otimes z)$

end *Körper.*

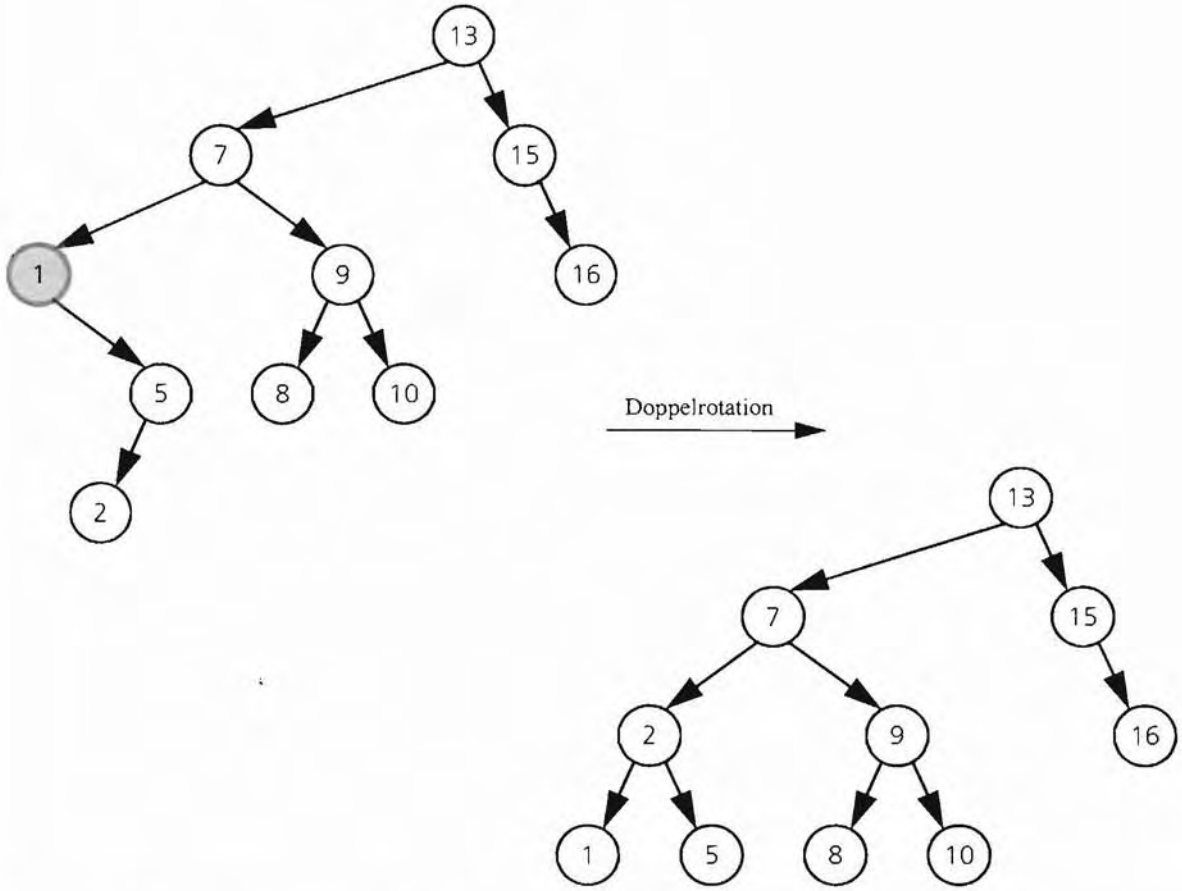
(b)

Zur algebraischen Spezifikation müsste man die Trägermenge und Funktionen direkt angeben. Allerdings ist weder K, noch $\oplus$, $\otimes$, n oder e in der Aufgabenstellung näher bestimmt. Jede entsprechende Festlegung (z.B. $K = \mathbb{R}$) würde aber eine Einschränkung bedeuten.

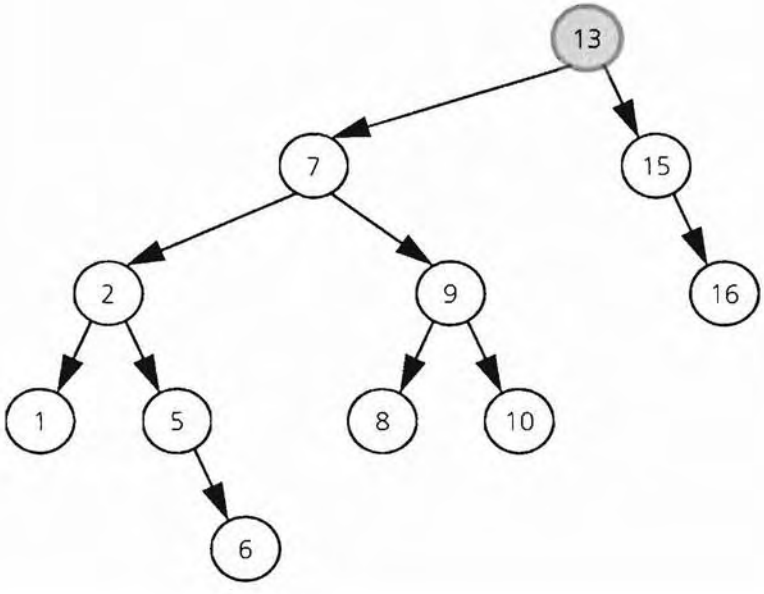
Aufgabe 2

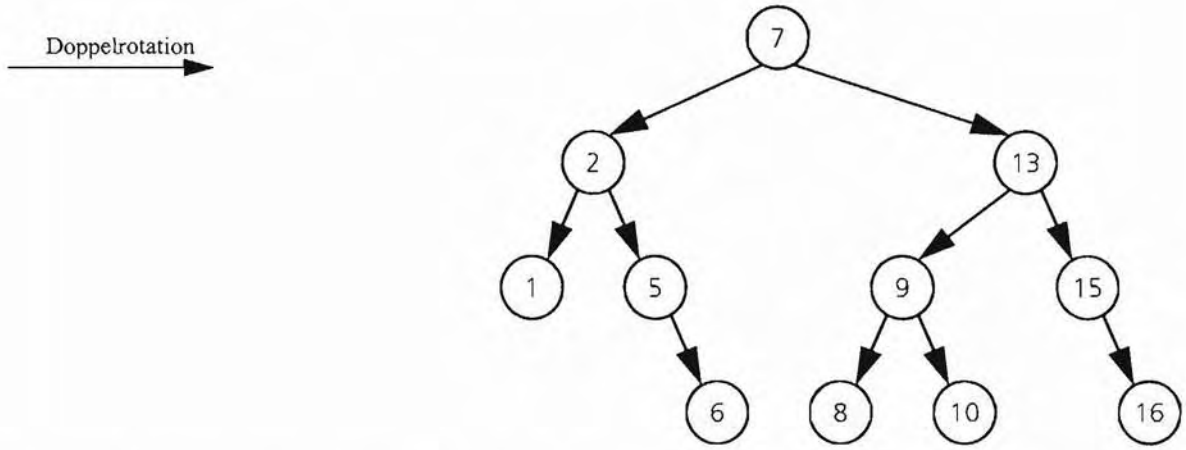
(a)

Beim Einfügen der 2 kommt es zu einer Verletzung der Balance am markierten Knoten:

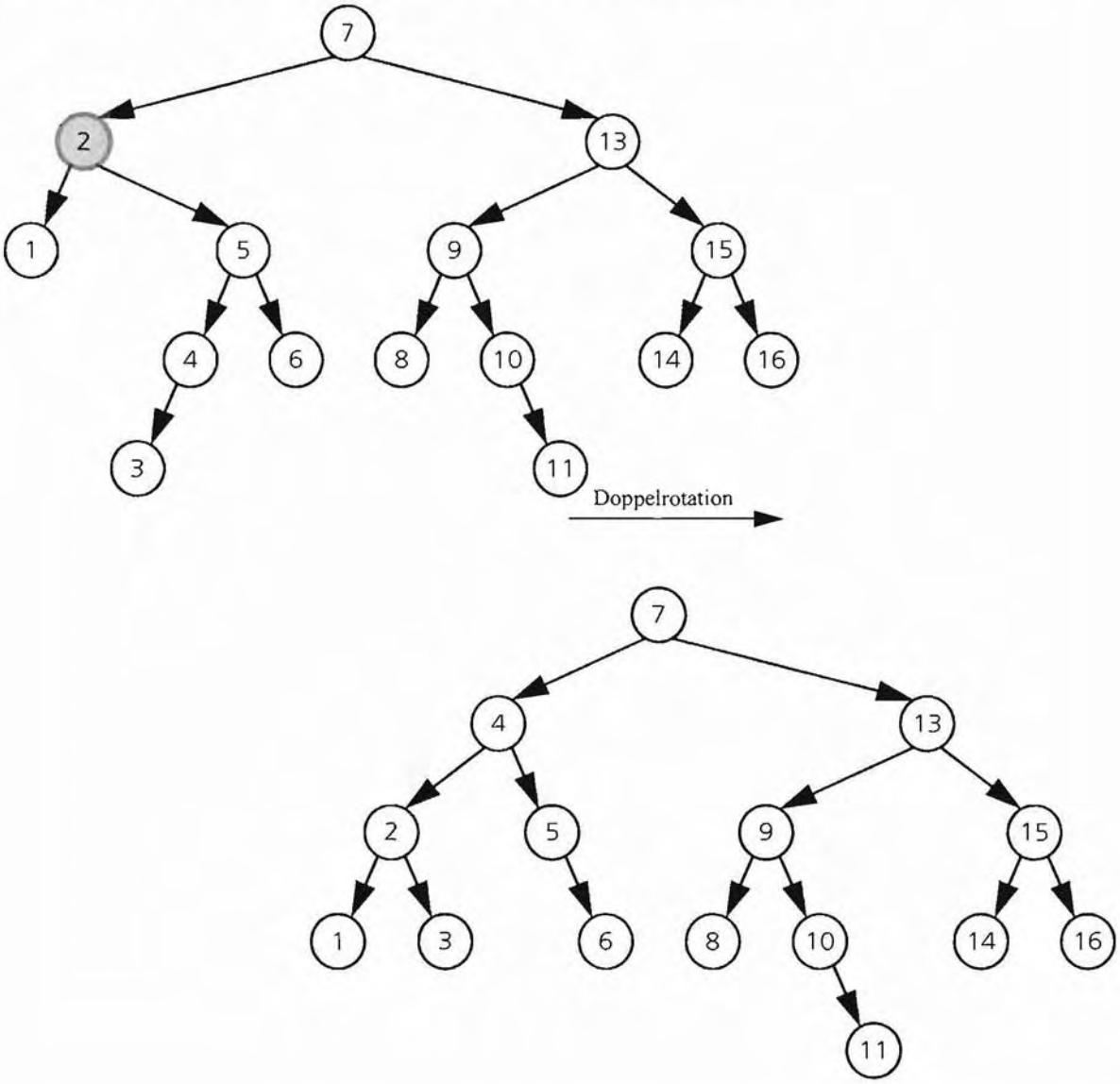


Das Balancekriterium wird beim Einfügen der 6 wieder verletzt:



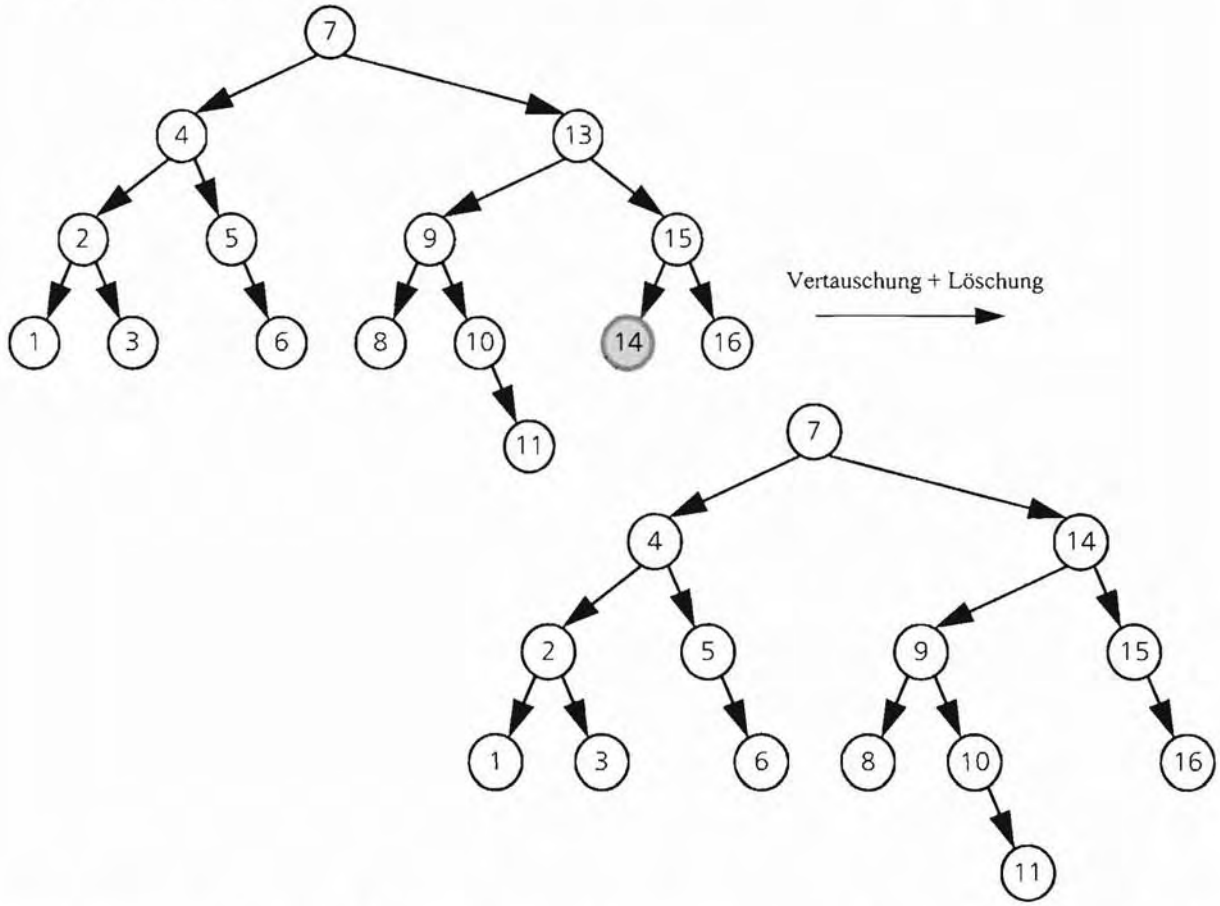


Die nächste Doppelrotation wird durch das Einfügen der 3 notwendig:

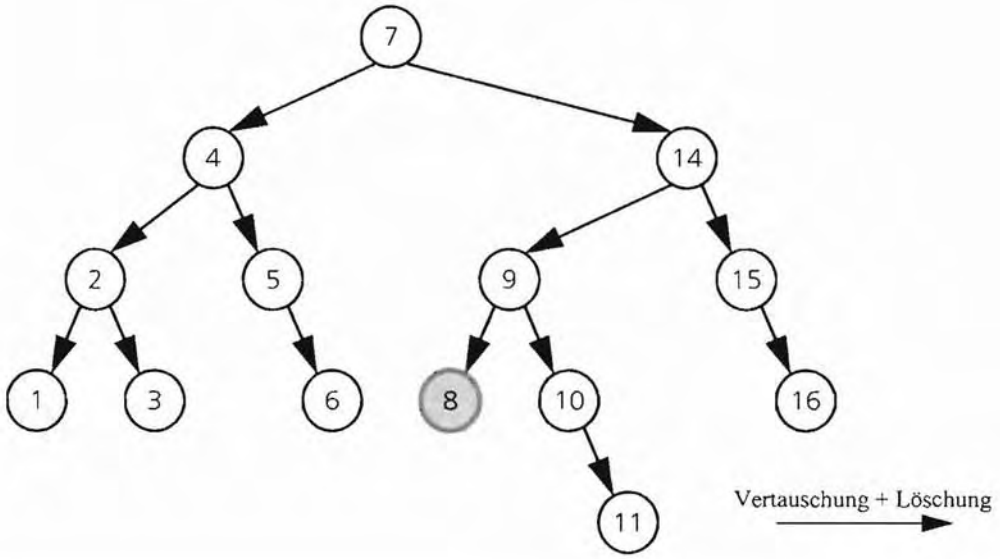


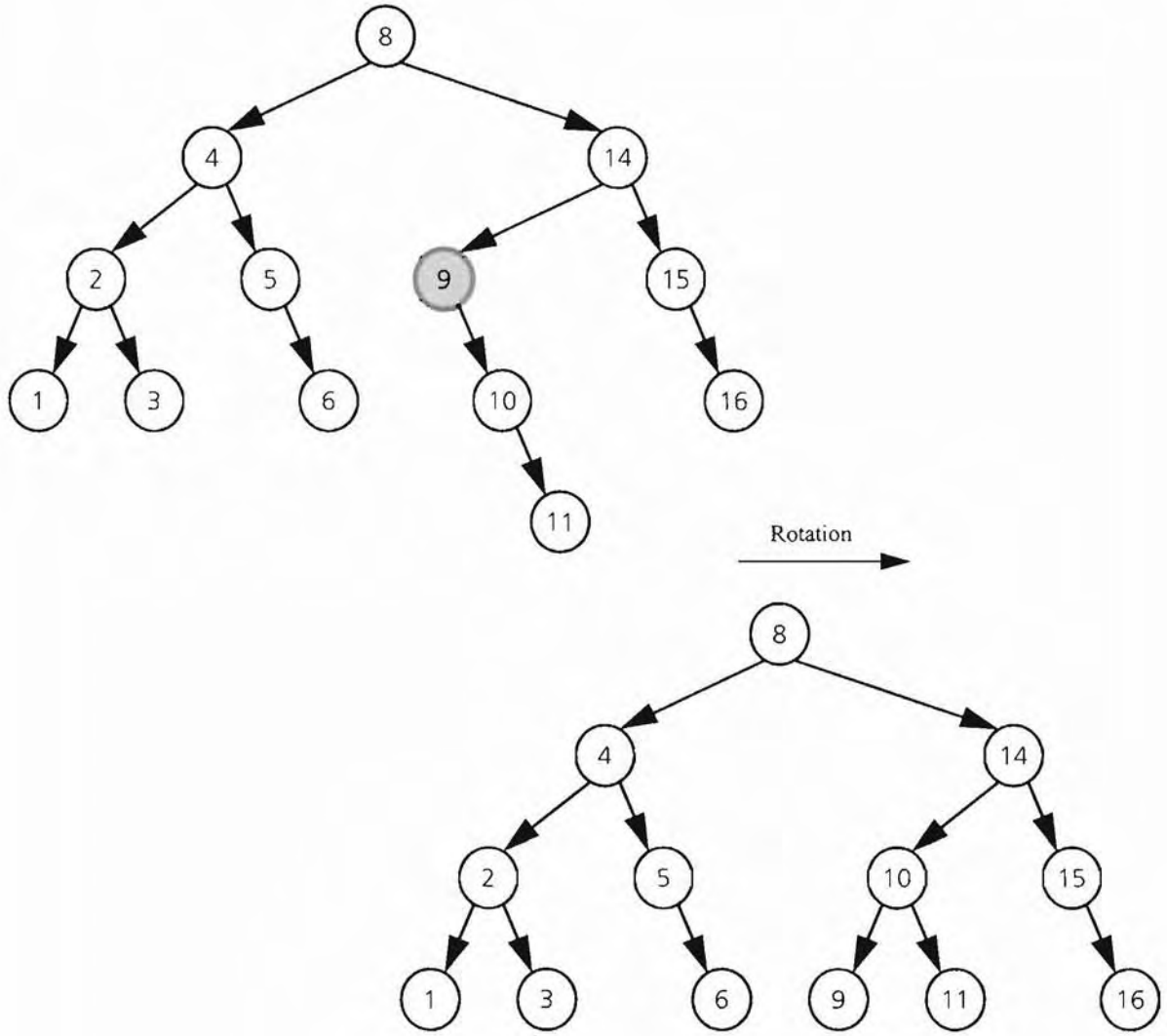
(b)

Das Löschen des Schlüssels 13 erfordert nur eine Vertauschung mit dem Knoten 14. Eine Rebalancierung ist nicht erforderlich.

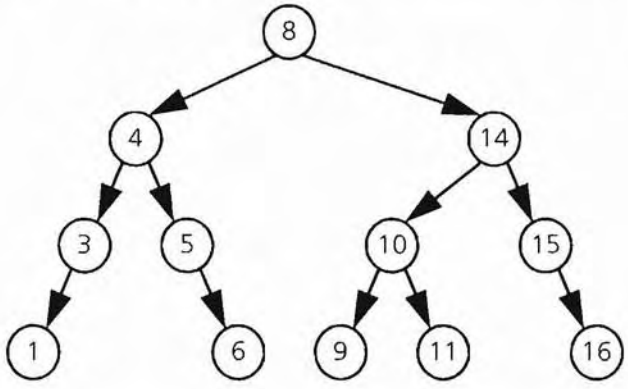


Zum Entfernen des Knotens 7 wird dieser zunächst durch Knoten 8 ersetzt. Anschließend erfolgt eine Rebalancierung:





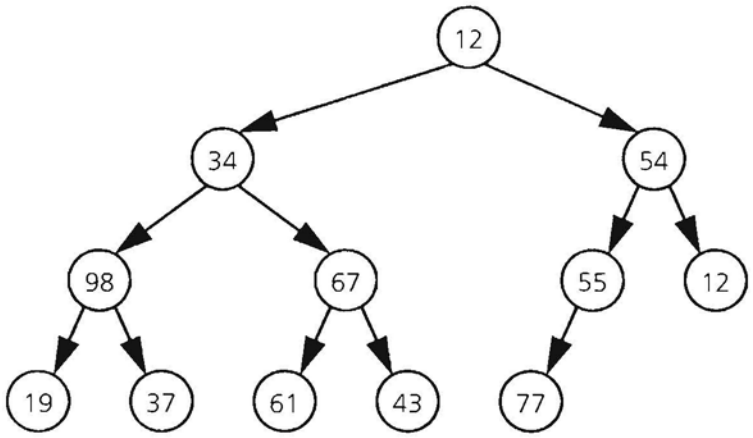
Abschließend wird Knoten 3 an die Stelle von 2 verschoben und Knoten 2 gelöscht. Als Ergebnisbaum ergibt sich folgender AVL-Baum:



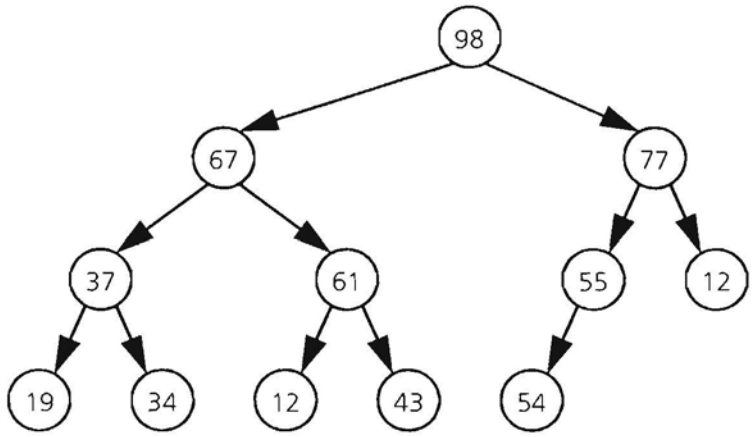
Aufgabe 3

(a)

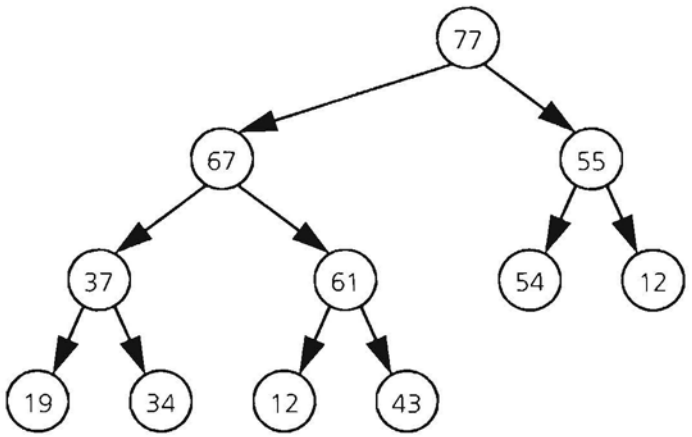
Die Baumeinbettung der Folge ist:



Nach Aufbau des Maximum-Heaps sieht der Baum wie folgt aus:



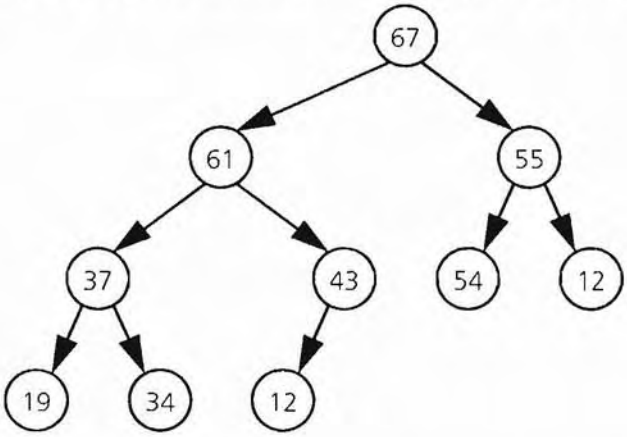
Die 98 wird in die sortierte Folge aufgenommen (mit der 54 getauscht), und anschließend wird die 54 einsinken gelassen. Wir erhalten:



F = 98

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

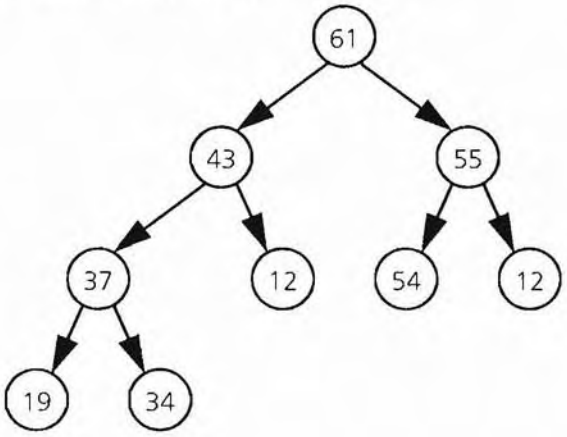
Nach Verarbeitung des jetzt maximalen Elements (77) ergibt sich folgendes Bild:



F = 77 - 98

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

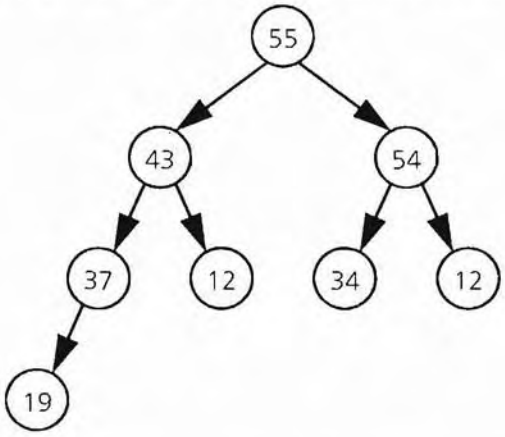
Durch das Einsinkenlassen der 12 erhält man:



F = 67 - 77 - 98

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

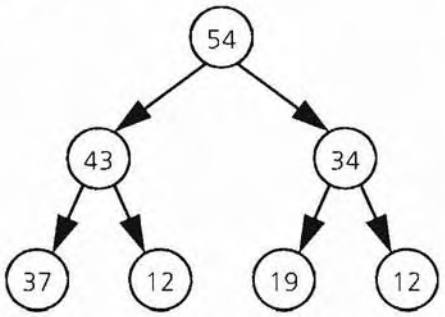
Nach dem nächsten Verarbeitungsschritt erhalten wir:



F = 61 - 67 - 77 - 98

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

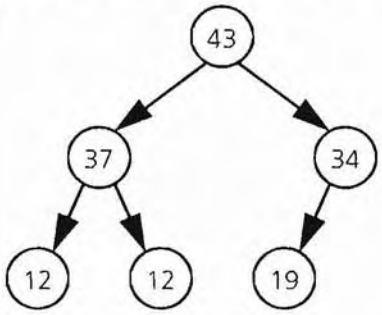
Danach ergibt sich:



F = 55 - 61 - 67 - 77 - 98

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

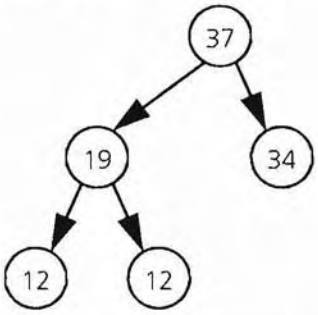
Durch Hinzufügen der 54 in die sortierte Folge und Einsinkenlassen der 12 erhält man:



F = 54 - 55 - 61 - 67 - 77 - 98

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

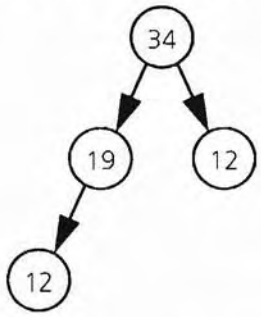
Das Einsinkenlassen der 19 ändert den Heap wie folgt:



F = 43 - 54 - 55 - 61 - 67 - 77 - 98

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

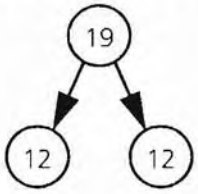
Nun wird die 37 zur sortierten Folge hinzugenommen:



F = 37 - 43 - 54 - 55 - 61 - 67 - 77 - 98

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

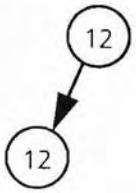
Im nächsten Schritt erhält man:



F = 34 - 37 - 43 - 54 - 55 - 61 - 67 - 77 - 98

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

Danach wird die 19 mit der 12 getauscht und die 12 einsinken gelassen:



F = 19 - 34 - 37 - 43 - 54 - 55 - 61 - 67 - 77 - 98

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

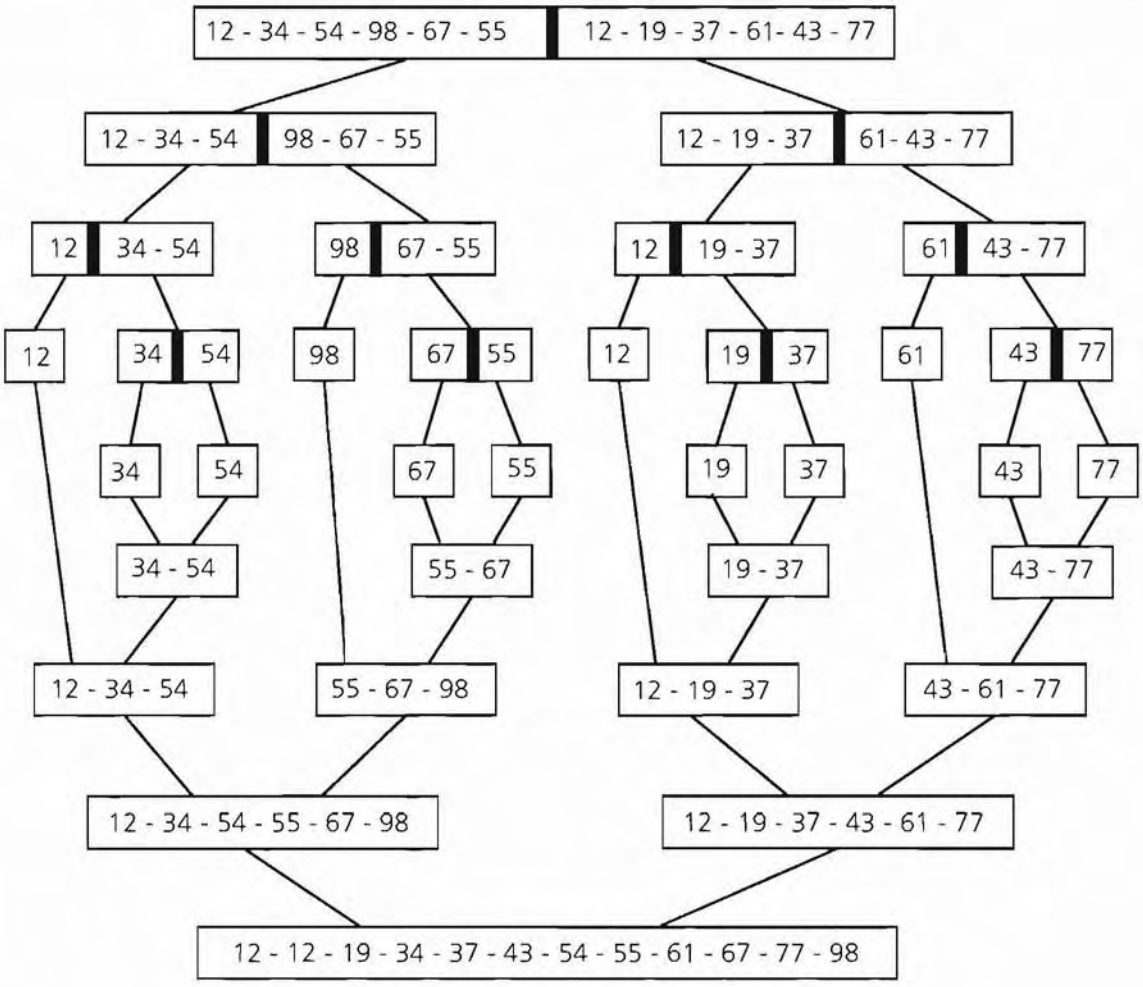
Nach dem letzten Schritt hat man schließlich die sortierte Folge:

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

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

(b)

Der Aufrufbaum von MergeSort ist:

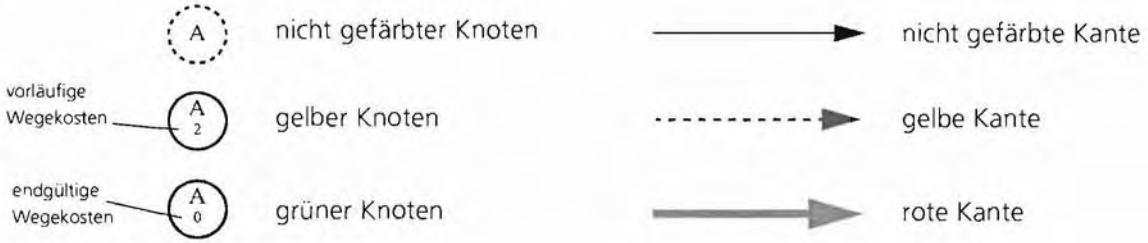


(c)

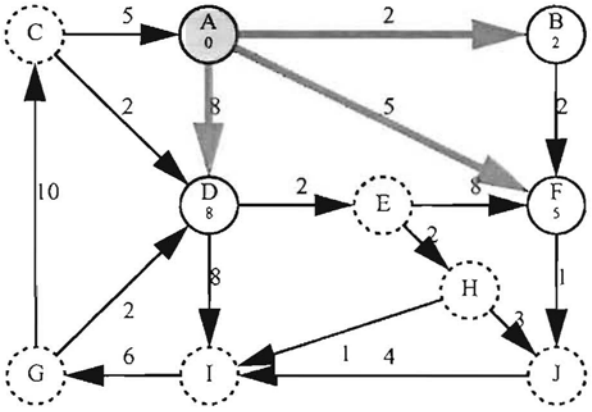
Diese Aufgabe erledigt der im Kurstext vorgestellte Divide-Schritt von QuickSort (*partition*), wenn man als Schlüsselwert 0 verwendet. Dieser hat lineare Laufzeit.

Aufgabe 4

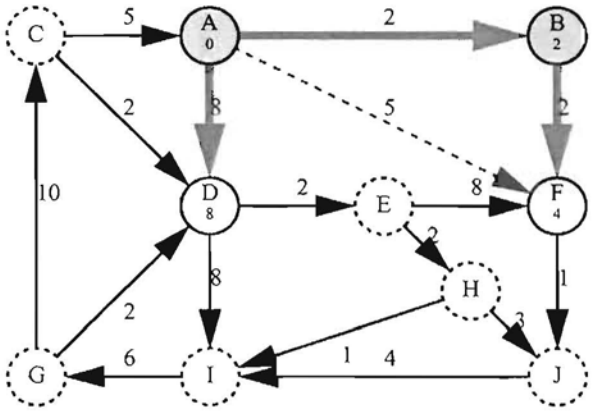
Wir verwenden die folgende Notation:



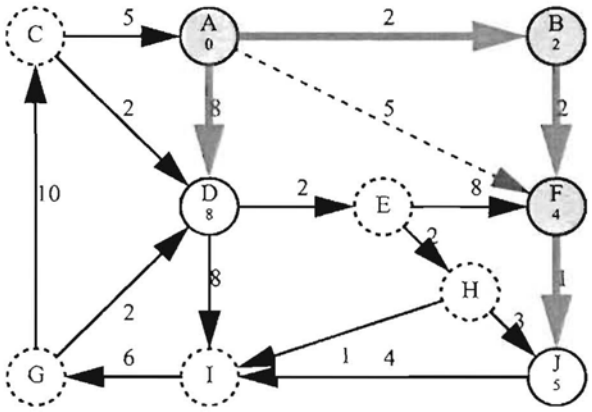
Wir beginnen mit Knoten *A*, der mit Wegekosten 0 gekennzeichnet wird. Von diesem Knoten ausgehend, werden seine Nachbarn in die Menge der gelben Knoten aufgenommen:



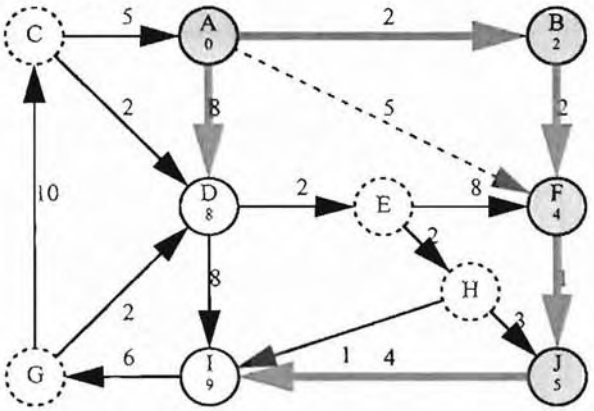
Nun wird der Knoten *B* grün gefärbt, was eine Aktualisierung der Kosten zum Knoten *F* bedeutet:



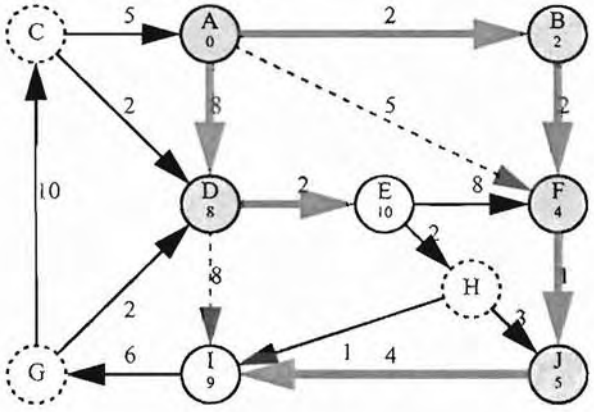
Dann wird Knoten *F* in die Menge der grünen Knoten aufgenommen:



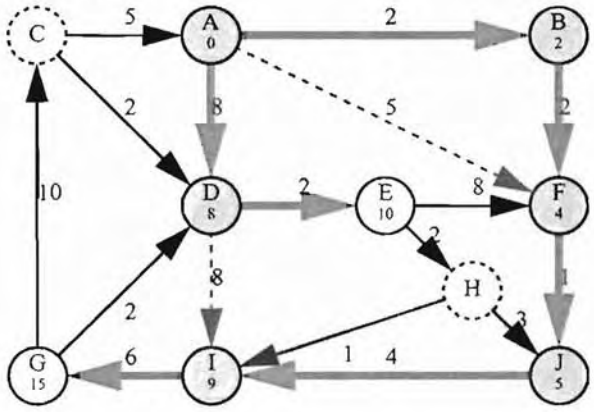
Da Knoten *J* nun innerhalb der Menge der gelben Knoten die geringste Kostenmarkierung aufweist, wird er als nächstes grün gefärbt:



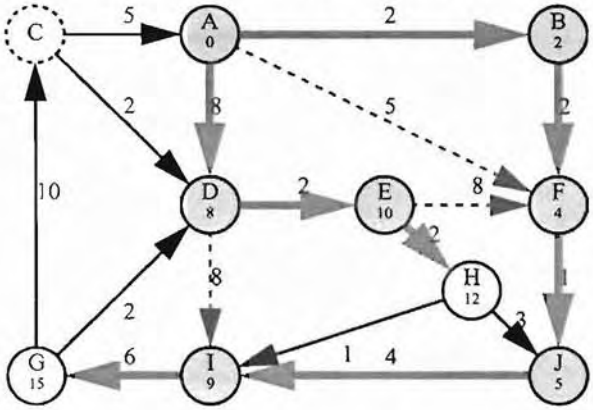
Da 8 < 9 gilt, wird nun Knoten *D* grün gefärbt:



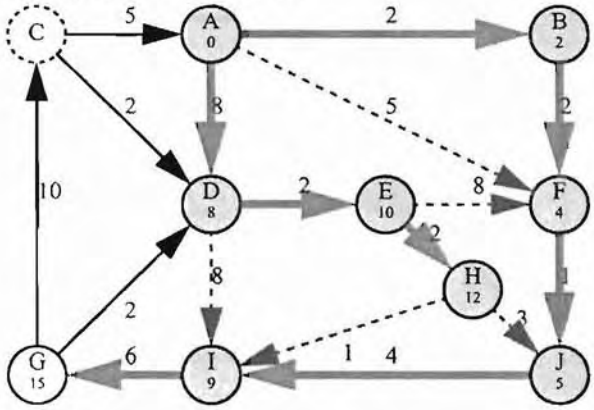
Knoten *I* ist der nächste Kandidat:



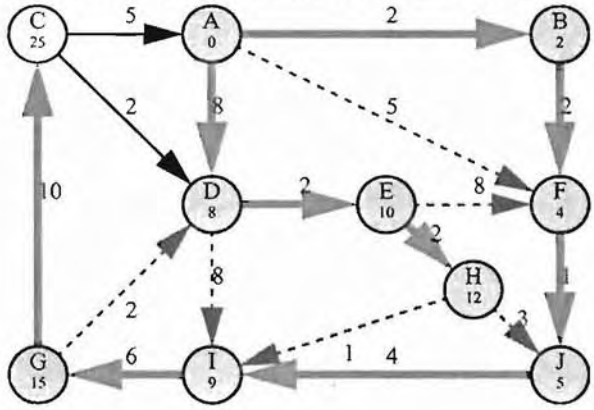
Knoten *E* hat jetzt die geringste Kostenmarkierung innerhalb der gelben Knoten:



Das gerade für Knoten *E* Gesagte, gilt nun für *H*:



Knoten *G* wird jetzt gewählt, da dieser der einzige gelbe Knoten ist:



Letztlich wird noch Knoten *C* grün gefärbt und alle gesuchten Wege sind gefunden ☺.