Algorithmen und Datenstrukturen
Klausur WS 2015/16

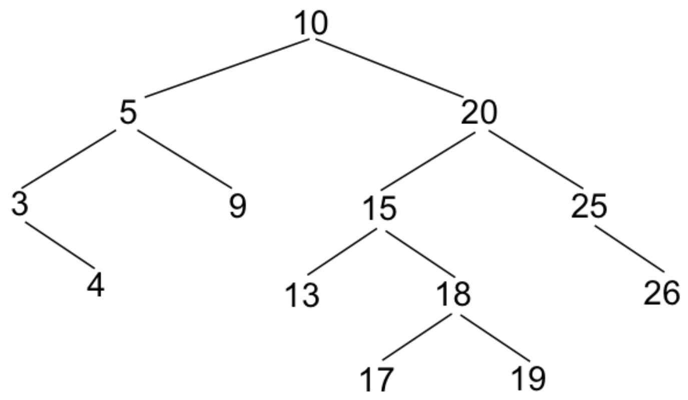
Angewandte Informatik Bachelor

Name	
Matrikelnummer	

Aufgabe 1	AVL-Baum	16	
Aufgabe 2	Algorithmus von Floyd	16	
Aufgabe 3	Heaps	12	
Aufgabe 4	Erreichbarkeit in einem Graphen	16	
Summe		**60**	

Aufgabe 1 AVL-Baum (16 Punkte)

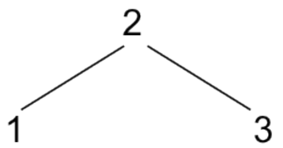
a) Gegeben ist folgender binärer Suchbaum. Der Tiefenunterschied zwischen Knoten 9 und 19 beträgt 2. Warum erfüllt der Baum trotzdem die AVL-Eigenschaft?



b) Löschen Sie in dem Baum aus a) den Knoten 10. Halten Sie dabei die folgende Regel ein: Wird ein Knoten mit zwei Kindern gelöscht, dann wird er durch das Minimum im rechten Teilbaum ersetzt.

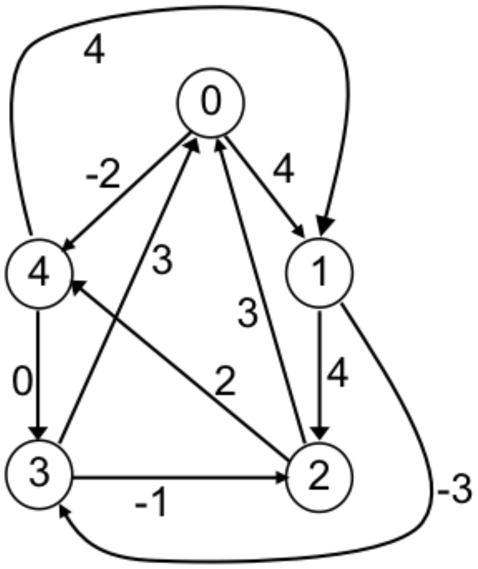
c) Löschen Sie in dem Baum aus a) den Knoten 9.

c) Fügen Sie in folgendem AVL-Baum nacheinander die Zahlen 4, 5, 6, 7, 8, 9, 10 ein.



Aufgabe 2 Algorithmus von Floyd (16 Punkte)

a) Berechnen Sie für folgenden gerichteten Graphen mit dem Algorithmus von Floyd für alle Knotenpaare einen günstigsten Weg. Es müssen sowohl die Distanzmatrizen D^k als auch die Vorgängermatrizen P^k berechnet werden (siehe nächste Seite). Heben Sie die Änderungen in D^k und P^k farblich hervor.



b) Was sind die Kosten für den günstigsten Weg von Knoten 3 nach Knoten 1?
Geben Sie an, wie sich der kürzeste Weg aus der Vorgängermatrix P^4 ergibt.

D^{-1}

0	4	∞	∞	−2
∞	0	4	−3	∞
3	∞	0	2	∞
3	∞	−1	0	∞
∞	4	∞	0	0

P^{-1}

−	0	−	−	0
−	−	1	1	−
2	−	−	2	−
3	−	3	−	−
−	4	−	4	−

D^0

P^0

D^1

P^1

D^2

P^2

D^3

P^3

D^4

P^4

Aufgabe 3 Heaps (12 Punkte)

a) Fügen Sie nacheinander die folgenden acht Zahlen 1, 2, 3, 4, 8, 7, 6, 5 in einen leeren binären Heap ein. Der Heap ist absteigend heap-geordnet: d.h. in heap[0] steht das Maximum und heap[i] ≥ heap[2i+1] und heap[i] ≥ heap[2i+2] (Eltern ≥ Kinder). Benutzen Sie eine graphische Darstellung der Heaps!

b) Fügen Sie nacheinander die folgenden zehn Zahlen 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 in einen leeren binomialen Heap ein. Die einzelnen Binomialbäume sind absteigend heap-geordnet. Benutzen Sie eine graphische Darstellung!

Aufgabe 4 Erreichbarkeit in einem Graphen (16 Punkte)

In einem ungerichteten Graphen G sind zwei Knoten u und v gegenseitig erreichbar, falls es einen Weg von u nach v gibt. Es sollen zwei Ansätze, die die Erreichbarkeit prüfen, miteinander verglichen werden.

a) Skizzieren Sie ein Verfahren, das mit einer Tiefensuche die gegenseitige Erreichbarkeit von zwei Knoten u und v prüft. Die Tiefensuche muß nicht beschrieben werden!

b) Skizzieren Sie ein Verfahren, das mit einer Union-Find-Struktur die gegenseitige Erreichbarkeit von zwei Knoten u und v prüft. Sie können voraussetzen, dass die Menge der Knoten V = {0, 1, 2, ..., n-1} ist. Die Union-Find-Struktur teilt die Menge V so in disjunkte Teilmengen auf, dass in einer Teilmenge alle diejenigen Knoten enthalten sind, die gegenseitig erreichbar sind.

Teilen Sie das Verfahren auf in eine Vorverarbeitung, in der die für einen Graphen G eine Union-Find-Struktur aufgebaut wird, und der eigentlichen Prüfung der Erreichbarkeit. Die Funktionen union(u, v) und find(v) dürfen als gegeben angenommen werden!

c) Geben Sie für die beiden Ansätze den Aufwand mit Hilfe der O-Notation an (in Abhängigkeit von Anzahl Knoten |V| bzw. Anzahl Kanten |E|).

Ansatz	Vorverarbeitung	Prüfen der Erreichbarkeit
Tiefensuche	-	
Union-Find-Struktur		