

Aufgaben zur Veranstaltung Algorithmen und Datenstrukturen, SS 2024

H. Pflug, J. Dietel

FH Aachen, Campus Jülich; IT Center, RWTH Aachen

Präsenzaufgaben 11

10./11.06.2024

Die Lösung der Aufgaben wird am Ende der Übung von Ihnen vorgestellt.

Aufgabe 1

Wenden Sie Dijkstras Algorithmus auf den folgenden Graphen an, indem Sie die jeweiligen Tabellen ausfüllen. Startknoten ist Knoten 1.

	1	2	3	4	5	6
1	0	--	14	--	6	4
2	7	0	2	4	3	7
3	--	3	0	11	--	10
4	3	1	--	0	--	18
5	--	16	4	--	0	--
6	15	--	7	--	15	0

V _i	d					p				
	2	3	4	5	6	2	3	4	5	6

Aufgabe 2

Heaps sind gut für eine Feldeinbettung geeignet, weil sie linksvollständig sind. Für binäre Suchbäume (und allgemein für Binärbäume) allgemein entstehen bei der Feldeinbettung Lücken. Dem kann man aber abhelfen, indem man für die Einbettung statt des Feldes eine Hashtabelle nimmt. Der Schlüssel der Hashtabelle entspricht dem Index des Feldes. Schreiben Sie eine Klasse BinaryHashSearchTree, die einen binären Suchbaum implementiert. Intern wird der binäre Suchbaum durch eine „Feld“-Einbettung in einer Hashtabelle realisiert. Implementieren Sie die unten angegebenen Methoden.

```
public class BinaryHashSearchTree<T extends Comparable> {  
    private HashMap<Integer, T> hash = new HashMap<>();  
  
    // Fuegt element zum Binaerbaum hinzu. Falls element  
    // schon vorhanden ist, passiert nichts  
    public void add(T element) { .. }  
  
    // Gibt true zurueck, falls element im Binaerbaum vorhanden ist,  
    // sonst false.  
    public boolean contains(T element) { .. }  
}
```

Aufgabe 3

Beantworten Sie folgende Fragen zu Aufgabe 2:

- a) Warum ist dieser Ansatz zum Löschen von Elementen ungeeignet? Begründen Sie Ihre Antwort anhand von Laufzeitkomplexitäten.
- b) Warum ist dieser Ansatz zur Umsetzung von AVL-Bäumen ungeeignet? Begründen Sie Ihre Antwort anhand von Laufzeitkomplexitäten.

Aufgabe 4

Schreiben Sie eine Funktion

```
public static int[][] difference(int[] a, int[] b)
```

Die Funktion gibt ein Feld von zwei int-Feldern zurück.

- Im ersten Feld stehen alle Zahlen, die im Feld a vorhanden sind, aber nicht im Feld b. Jede Zahl ist dabei nur einmal vorhanden.
- Im zweiten Feld stehen alle Zahlen, die im Feld b vorhanden sind, aber nicht im Feld a. Jede Zahl ist dabei nur einmal vorhanden.

Die Funktion soll eine Laufzeit haben, die maximal $O(n_a+n_b)$ ist, wobei n_a und n_b die Anzahl der Elemente in a und b bezeichnet.