

Übungen zur Vorlesung
Algorithmen und Datenstrukturen
WiSe 2019/20
Blatt 7

Wichtige Hinweise:

- > Falls Sie bei der Bearbeitung einer Aufgabe größere Schwierigkeiten hatten und deswegen die Bearbeitung abgebrochen haben, so versuchen Sie bitte Ihre Schwierigkeiten in Form von Fragen festzuhalten. Bringen Sie Ihre Fragen einfach zur Vorlesung oder zur Übung mit!
- > Kursraum: <https://elearning.uni-regensburg.de/course/view.php?id=9228>

Aufgabe 1:

Implementieren Sie eine Datenstruktur für einen binären verketteten Suchbaum in C, C++, C# oder Java und realisieren Sie insbesondere eine Funktion zum Löschen eines Elements aus einem binären verketteten Suchbaum (sowohl iterativ als auch rekursiv). Geben Sie die Laufzeit für Ihre implementierten Funktionen in Abhängigkeit der in dem Baum vorhandenen n Elemente an.

Aufgabe 2:

Implementieren Sie einen Algorithmus, der bei Eingabe der Preorder- und Inorder-Darstellung oder der Inorder- und Postorder-Darstellung eines binären Baums den Baum eindeutig rekonstruiert. Entwickeln Sie einen weiteren Algorithmus, der bei Eingabe der Preorder-Darstellung eines binären, verketteten Suchbaums den binären, verketteten Suchbaum eindeutig rekonstruiert.

Aufgabe 3:

Geben Sie die AVL-Bäume an, die entstehen, wenn Sie die Werte 5, 6, 9, 12, 13, 3, 8, 10, 11, 16, 15, 14, 4, 2, 1 in genannter Reihenfolge einfügen. Geben Sie anschließend die AVL-Bäume an, die entstehen, wenn Sie die Werte 12, 8, 5, 4, 3, 6, 15, 14 in genannter Reihenfolge entfernen. Überprüfen Sie Ihre AVL-Bäume mit Hilfe geeigneter Implementierungen (implementieren Sie insbesondere die Funktion `Delete(object o)`)!

Aufgabe 4:

Konstruieren Sie einen minimalen AVL-Baum der Höhe 5. Wie viele minimale Bäume der Höhe 5 gibt es? Entwerfen Sie einen Algorithmus, der alle minimalen AVL-Bäume der Höhe h ausgibt und berechnet, wie viele solcher Bäume es gibt.