

Datenstrukturen und Algorithmen

Übung 9 – Suchbäume

Aufgabe 9.1 – Explizite Suchbäume [30 Punkte]

Gegeben Sei die folgende Sequenz.

3, 1, 5, 4, 6, 2, 7, 10

Fügen Sie die Elemente in einen leeren Suchbaum ein und zwar

(a) sukzessiv. 10 Punkte

(b) so, dass der Baum maximale Höhe hat. Geben Sie dazu auch die Reihenfolge der Einfügungen an.

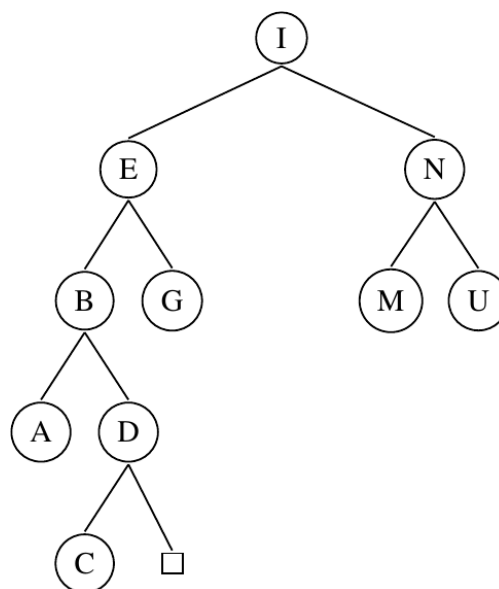
10 Punkte

(c) so, dass der Baum minimale Höhe hat. Geben Sie dazu auch die Reihenfolge der Einfügungen an.

10 Punkte

Aufgabe 9.2 – Optimierte Suchbäume [40 Punkte]

Gegeben Sei der folgende Suchbaum.



Die Folgenden Aufgaben sollen jeweils mit dem oben gegebenen Suchbaum bearbeitet werden, es soll nicht die Lösung des vorherigen Aufgabenteils weiterverwendet werden.

- (a) Löschen Sie **nacheinander** die Knoten A, I, G (der Baum soll danach 7 Knoten haben)
- (b) Ist der ursprüngliche Baum rot-schwarz färbbar? Wenn ja geben Sie eine solche Färbung an.
- (c) Rotieren Sie nach dem AVL-Schema der Vorlesung den Teilbaum mit Wurzel E. Geben Sie vor und nach dem Rotieren den Balance-Wert für jeden Knoten aus und beantworten Sie die Frage, ob der Baum vor und nach dem Rotieren die AVL-Eigenschaft erfüllt.

Hinweis: Der Balance Wert $b(v)$ für einen Knoten v mit linken Nachfolger l und rechtem Nachfolger r ist

$$b(v) = |h(l) - h(r)|$$

Wobei $h(u)$ die Höhe des Teilbaumes u zurückgibt.

Aufgabe 9.3 – Ausgeglichene Suchbäume _____ [30 Punkte]

- (a) Beschreiben Sie warum in einem Rot-Schwarz Baum ein schwarzer Knoten nie ein Blatt als Nachbarn haben kann 10 Punkte
- (b) Beschreiben Sie wie man einen ausgeglichenen Suchbaum als Prioritätswarteschlange verwenden kann. Geben Sie dazu die Laufzeiten der einzelnen Operationen an. Welchen Vorteil könnte ein ausgeglichener Suchbaum zu einem Heap haben?

Hinweis: Wer Probleme bei dieser Aufgabe hat kann sich den *Completely Fair Scheduler* ansehen, der im Linux Kernel verwendet wird und einen Rot-Schwarz Baum zum Scheduling verwendet.

20 Punkte

AVL Rebalanzierungsregeln

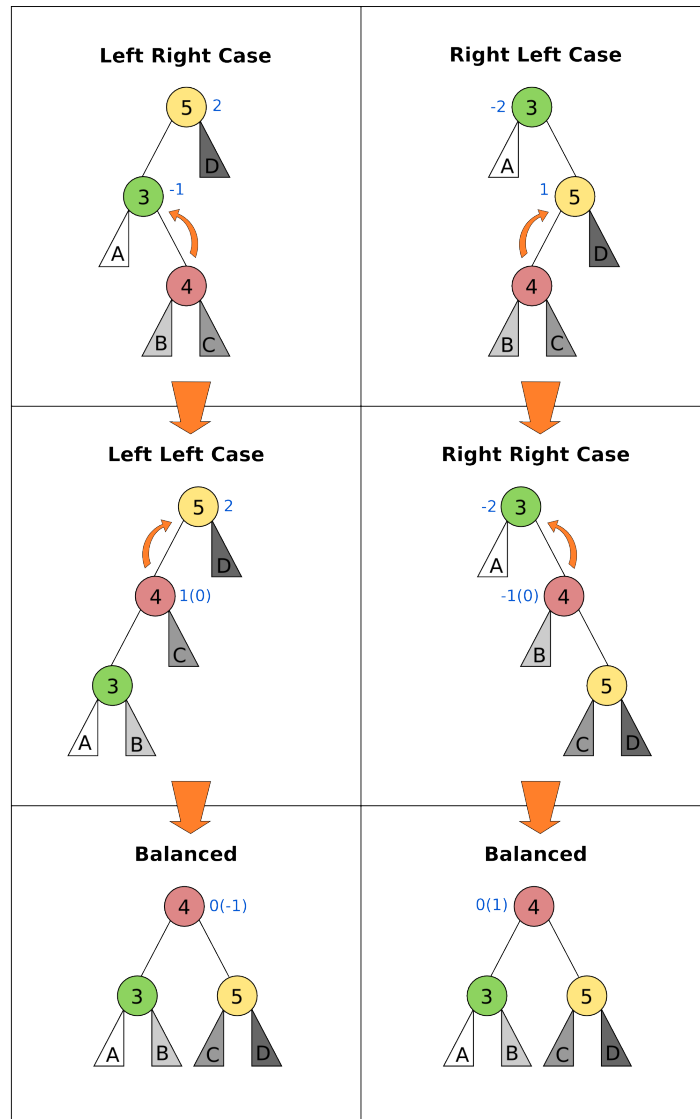


Abbildung 1: AVL Rebalanzierung. Quelle: http://en.wikipedia.org/wiki/AVL_tree