

10. Übungsblatt

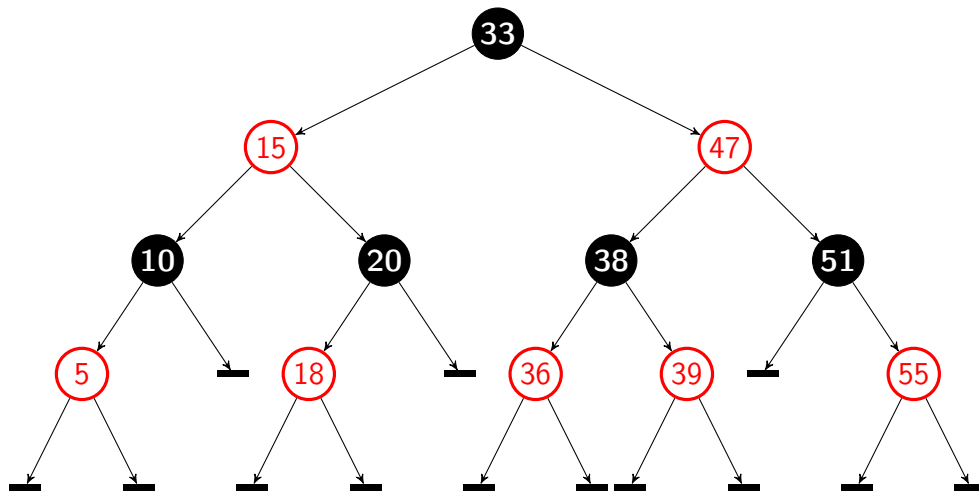
zur Vorlesung

Grundzüge der Informatik I

Abgabe über Ilias bis zum 21.6. 14:00 Uhr.
Besprechung in Kalenderwoche 26.

Aufgabe 1 Rot-Schwarz-Bäume (2 + 5 + 5 Punkte)

- Vervollständigen Sie die Methode RS-Einfügen-Fix aus der Vorlesung 17 (Folie 28), indem Sie den else Fall in Zeile 10 ergänzen.
- Fügen Sie die Schlüssel 30, 42, 41, 50, 55, 1, 3 in dieser Reihenfolge in einen anfangs leeren Rot-Schwarz-Baum ein und notieren Sie dabei nach jeder Operation den resultierenden Baum.
- Betrachten Sie den folgenden Rot-Schwarz-Baum.



Löschen Sie die Werte 5, 10, 47, 20 und 18 in der genannten Reihenfolge. Zeichnen Sie den Rot-Schwarz-Baum nach jedem Löschen und geben Sie an, welche Fälle durchgeführt wurden.

Aufgabe 2 *Suchbäume* (4 + 2 + 2 Punkte)

Sei T ein binärer Baum, der nicht zwingend die Suchbaumeigenschaft erfüllt. Es soll nun geprüft werden, ob T ein Suchbaum ist. Für den binären Baum T liefert $\text{root}(T)$ den Wurzelknoten von T . Für jeden Knoten $v \in V$ im binären Baum T stehen folgende Operationen mit Laufzeit $\mathcal{O}(1)$ zur Verfügung.

- $\text{key}(v)$ liefert den im Knoten v enthaltenen Schlüssel.
- $\text{lc}(v)$ liefert einen Zeiger auf das linke Kind von v in T . Hat v kein linkes Kind, liefert $\text{lc}(v)$ den Wert NIL.
- $\text{rc}(v)$ liefert einen Zeiger auf das rechte Kind von v in T . Hat v kein rechtes Kind, liefert $\text{rc}(v)$ den Wert NIL..
- $\text{leaf}(v)$ liefert TRUE, falls der Knoten v in T ein Blatt ist. Sonst liefert $\text{leaf}(v)$ den Wert FALSE.

Für diese Aufgabe soll T ein *vollständiger* binärer Baum sein. Das heißt, dass jeder innere Knoten in T genau zwei Kinder hat und alle Blätter von T auf der selben Höhe sind.

- a) Entwerfen Sie einen Teile-und-Herrsche-Algorithmus, der bei Eingabe eines binären Baumes T mit Knotenmenge V den Wert TRUE zurückgibt, falls T die Suchbaumeigenschaft erfüllt. Erfüllt T die Suchbaumeigenschaft nicht, soll der Algorithmus FALSE zurückgeben. Beschreiben Sie Ihren Algorithmus in eigenen Worten und geben Sie den Algorithmus auch in Pseudocode an. Für die volle Punktzahl wird ein Algorithmus erwartet, dessen Laufzeit durch $\mathcal{O}(|V|)$ beschränkt ist.
- b) Analysieren Sie die Laufzeit Ihres Algorithmus.
- c) Beweisen Sie die Korrektheit Ihres Algorithmus.