



10. ÜBUNG ZUR VORLESUNG PROGRAMMIERUNG II, SS 2004

Aufgabe 1: Rekurrenz, Substitutionsmethode (10 Punkte)

Lösen Sie die folgenden Rekurrenzgleichungen mit der *Substitutionsmethode*.

- i) $T(n) = 3T(\lfloor \frac{n}{2} \rfloor) + n^2$
 - ii) $T(n) = 3T(\lfloor \frac{n}{8} \rfloor) + T(\lfloor \frac{n}{4} \rfloor) + 1$
 - iii) $T(n) = 2T(\lfloor \frac{n}{4} \rfloor) + 8T(\lfloor \frac{n}{16} \rfloor) + n$
-

Aufgabe 2: Rekurrenz, Mastertheorem (10 Punkte)

Welcher der folgenden Rekurrenzen können Sie mit dem *Master-Theorem* auswerten? Begründen Sie Ihre Antwort und bestimmen Sie in den Fällen, in denen das Theorem anwendbar ist, die Lösung der Gleichung.

- i) $T(n) = 49\sqrt{7}T(\frac{n}{7}) + 7\sqrt{n^5} + 49\sqrt{7}$
 - ii) $T(n) = 27T(\frac{n}{3}) + \frac{n^3}{\log_3(n)}$
 - iii) $T(n) = 9T(\frac{n}{3}) + 3\log(n)n^2$
 - iv) $T(n) = 15625T(\frac{n}{5}) + \frac{1}{37}n^7$
 - v) $T(n) = 1296T(\frac{n}{6}) + n^3\sqrt{n}\log(n)$
-

Aufgabe 3: Rot-Schwarz-Baum (10 Punkte)

- i) Zeichnen Sie den Rot-Schwarz-Baum, der sich ergibt, wenn Sie der Reihe nach die Schlüssel $\{31, 15, 36, 41, 34, 43\}$ in einen anfangs leeren Baum einfügen.
- ii) Entfernen Sie nun der Reihe nach die Schlüssel $\{43, 41, 36, 15, 31, 34\}$ aus dem Baum und zeichnen Sie nach jedem Schritt, den sich ergebenden Rot-Schwarz-Baum.
- iii) Löschen Sie aus dem Rot-Schwarz-Baum von Abb.1 den Schlüssel 17 und zeichnen Sie den sich ergebenden Rot-Schwarz-Baum.

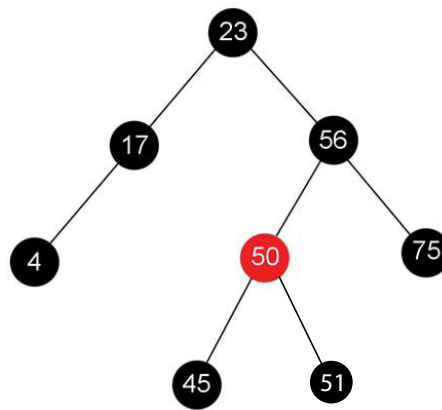


Abbildung 1: Rot-Schwarz-Baum von Aufgabe 3.

Aufgabe 4: Binäre Suchbäume (10 Punkte)

Die in der Vorlesung vorgestellte Implementierung von binären Suchbäumen ist nicht darauf ausgelegt, viele Elemente mit identischen Schlüsseln abzuspeichern.

- Bestimmen Sie die asymptotische Gesamtlaufzeit einer Sequenz von n `tree_insert`-Aufrufen, mit denen n Elemente mit identischem Schlüssel in einen anfangs leeren binären Suchbaum eingefügt werden.
- Um dieses Problem zu lösen kann man in der Implementierung von `tree_insert` den Fall `x->key == z->key` in Zeile 7 sowie den Fall `z->key == p->key` in Zeile 13 abfragen und separat behandeln. In diesem Fall soll eine der drei folgenden Strategien verwendet werden:
 - i) Jeder Knoten erhält eine `bool` Variable `links`, die mit `true` initialisiert wird und jedes Mal, wenn der Knoten während einem Aufruf von `tree_insert` besucht wird, logisch negiert wird. Hat `links` den Wert `true`, wird in Zeile 7 `x->left_child` gewählt, ansonsten `x->right_child`.
 - ii) Jeder Knoten speichert eine Liste mit Elementen mit gleichen Schlüsseln, an die `z` einfach angehängt wird.
 - iii) `x` wird zufällig auf `x->right_child` oder `x->left_child` gesetzt.

Geben Sie für die Strategien (i) und (ii) die asymptotische Gesamtlaufzeit einer wie oben definierten Sequenz von n `tree_insert` Aufrufen an. Bestimmen Sie für Strategie (iii) die worst-case Laufzeit und geben Sie eine informelle Herleitung für die durchschnittliche Laufzeit dieser Strategie an.

Abgabe: 02.07.2004 (Vor der Vorlesung)