

Übungsblatt 8

Aufgabe 8.1

4+2 Punkte

- Fügen Sie die Schlüssel $1, \dots, 10$ in dieser Reihenfolge in einen zu Beginn leeren B -Baum der Ordnung $t = 2$ ein. Zeichnen Sie den Baum nach jeder Operation.
- Löschen Sie anschließend die Schlüssel $8, 4, 9$ in dieser Reihenfolge. Bevorzugen Sie bei Rotationen und Verschmelzungen den linken Nachbarn vor dem rechten Nachbarn. Zeichnen Sie den Baum nach jeder Operation.

Hinweis: Die Dateien `Ue8_A1a.txt`, `Ue8_A1b.txt` aus Aufgabe 8.3 enthalten die in Aufgabenteil (a) und (b) verwendeten Sequenzen. Sie können die dort angegebenen Ausgaben von `PREORDER` auf diesen Sequenzen heranziehen, um die Korrektheit Ihrer Lösung zu überprüfen.

Aufgabe 8.2

3+3 Punkte

Wir betrachten geschlossenes Hashing mit einer Hashtabelle der Größe $m = 13$. Fügen Sie die Schlüssel

1207, 286, 1195, 298, 62, 573, 623, 6280, 89, 199, 447

in der angegebenen Reihenfolge gemäß der Hashfunktion $h(x) := x \bmod m$ in die Hashtabelle ein. (Es genügt, das Endergebnis aller Einfügeoperationen anzugeben.) Ermitteln Sie die durchschnittliche und die maximale Anzahl von Sondierungen, die beim Einfügen eines Schlüssels nötig waren. Betrachten Sie als Strategien

- Lineares Sondieren
- Quadratisches Sondieren

Aufgabe 8.3

6 Punkte

Um die Korrektheit einer B -Baum-Implementierung empirisch zu überprüfen, betrachten wir den folgenden Test für einen nichtleeren B -Baum.

PREORDER(v)

- Gib die Anzahl ℓ der in v gespeicherten Schlüssel aus.
- if** (v ist Blatt) **then return**
- Bestimme die Kinder $u_1, \dots, u_{\ell+1}$ von v .
- for** $i = 1, \dots, \ell + 1$ **do** Rufe `PREORDER`(u_i) auf.

Der initiale Aufruf ist `PREORDER(root)`, wobei `root` die Wurzel des B -Baumes ist.

Implementieren Sie die Datenstruktur B -Baum, wie sie in der Vorlesung beschrieben wurde. Bevorzugen Sie bei Rotationen und Verschmelzungen, die beim Löschen auftreten, den linken Nachbarn vor dem rechten Nachbarn. Sie können die Korrektheit Ihrer Implementierung mittels obigem Test an den Instanzen `Ue8_A1a.txt`, `Ue8_A1b.txt` und `Ue8_sample.txt` testen, die auf der Homepage heruntergeladen werden kann. Die Ausgaben sollte wie folgt lauten:

<code>Ue8_A1a.txt</code>	1, 1, 1, 1, 2, 1, 1, 2
<code>Ue8_A1b.txt</code>	2, 2, 1, 2
<code>Ue8_sample.txt</code>	2, 4, 3, 5

Geben Sie die Ausgabe des Tests auf der Instanz `Ue8_exercise.txt` an. Das Dateiformat der Dateien ist ebenfalls auf der Homepage erklärt.

Aufgabe 8.4

2+2+2 Punkte

- (a) Wir werfen einen roten und einen blauen Spielwürfel. Bestimmen Sie die Wahrscheinlichkeit, dass der rote Würfel mindestens eine 3 und der blaue Würfel eine gerade Zahl zeigt.
- (b) Wir werfen einen roten und einen blauen Spielwürfel. Bestimmen Sie die Wahrscheinlichkeit, dass einer der beiden Würfel mindestens eine 3 und der andere Würfel eine gerade Zahl zeigt.
- (c) Wir werfen n -mal hintereinander einen gezinkten Würfel, bei dem die Wahrscheinlichkeit, eine Zahl i zu würfeln, gleich p_i ist. Bestimmen Sie, wie oft im Erwartungswert eine Primzahl gewürfelt wird.

Aufgabe 8.5

5+1 Zusatzpunkte

Ein B^* -Baum der Ordnung t ist ein B -Baum der Ordnung t , bei dem jeder Knoten außer der Wurzel und den Söhnen der Wurzel nicht nur mindestens $t - 1$ Schlüssel, sondern mindestens $\lfloor \frac{4t-2}{3} \rfloor$ Schlüssel enthält.

- (a) Beschreiben Sie eine INSERT-Operation für B^* -Bäume. Konzentrieren Sie sich dabei auf die wesentliche Idee. Quelltext ist nicht nötig.
- (b) Warum kann man nicht sicherstellen, dass die Söhne der Wurzel mindestens $\lfloor \frac{4t-2}{3} \rfloor$ Schlüssel besitzen?