10. Aufgabenblatt zur Vorlesung

Algorithmen und Programmierung III

Wolfgang Mulzer WiSe 2013/14

Abgabe am 9. Januar 2014 vor der Vorlesung in die jeweiligen Tutorenfächer

Dieser Zettel enthält nur Zusatzaufgaben. Das AlP-3-Team wünscht frohe Weihnachten und ein gutes neues Jahr.

Aufgabe 1 Rot-Schwarz Bäume und $(2, 4)$-Bäume *5 Punkte*

Ein *rot-schwarz Baum* ist ein binärer Suchbaum, in dem jeder Knoten eine Farbe hat: rot oder schwarz. Dabei muss folgendes gelten: (i) die Wurzel ist schwarz; (ii) die $\perp$-Knoten (d.h., die NULL-Knoten) sind schwarz; (iii) die Kinder eines roten Knoten sind schwarz; (iv) die *schwarze Tiefe* aller $\perp$-Knoten ist gleich, d.h., für alle $\perp$-Knoten ist die Anzahl der schwarzen Knoten auf dem Pfad von der Wurzel zum jeweiligen Knoten gleich.

Zeigen Sie: Rot-schwarz Bäume und $(2, 4)$-Bäume sind äquivalent. Genauer: es gibt eine lokale Transformation, welche Gruppen von Knoten im rot-schwarz Baum in Knoten im $(2, 4)$-Baum überführt, und umgekehrt. Geben Sie eine solche Transformation an, und begründen Sie, dass Ihre Transformation die Bedingungen an rot-schwarz Bäume und an $(2, 4)$-Bäume erfüllt.

Aufgabe 2 Huffman-Implementierung — Entwurf *10 Punkte*

Implementieren Sie den Huffman-Kodierungsalgorithmus. Genauer: Schreiben Sie zwei Java-Programme `HuffmanEncode` und `HuffmanDecode`. Bei Eingabe des Befehls `java HuffmanEncode foo` soll die Datei `foo` eingelesen und mit dem Huffman-Algorithmus (auf Byteebene) komprimiert werden. Das Ergebnis (zusammen mit der Kodetabelle) soll in eine Datei `foo.hc` geschrieben werden. Beim Aufruf von `java HuffmanDecode foo.hc` soll die komprimierte Datei `foo.hc` gelesen und die entkomprimierte Version in die Datei `foo.uc` geschrieben werden.

Überlegen Sie sich einen geeigneten Entwurf für das Programm. Welche Teilprobleme müssen gelöst werden? Welche abstrakten Datentypen werden benötigt? Wie lassen sich die Teilprobleme auf Klassen aufteilen? Wie sollen die Klassen miteinander interagieren? Wie soll der Programmablauf stattfinden? Beschreiben Sie knapp Ihren Entwurf und visualisieren Sie ihn mit geeigneten Diagrammen.

Aufgabe 3 Huffman-Implementierung — Umsetzung *15 Punkte*

Setzen Sie Ihren Entwurf aus der letzten Aufgabe in die Tat um und testen Sie Ihr Programm an geeigneten Beispielen.

Aufgabe 4 Erwartete Pfadlänge in einem binären Suchbaum *10 Punkte*

Seien n verschiedene Schlüssel aus einem total geordneten Universum gegeben. Der *zufällig aufgebaute binäre Suchbaum* auf diesen n Schlüsseln ist der binäre Suchbaum, welcher entsteht, wenn man die n Schlüssel in zufälliger Reihenfolge in einen anfangs leeren binären Suchbaum einfügt. Dabei ist jede der $n!$ möglichen Einfügereihenfolgen gleich wahrscheinlich.

(a) Zeigen Sie: die Wahrscheinlichkeitsverteilung, wenn man einen binären Suchbaum aus n Schlüsseln zufällig aufgebaut, unterscheidet sich von der Verteilung, wenn man gleichverteilt einen binären Suchbaum auf n Elementen auswählt.

Hinweis: Betrachten Sie den Fall $n = 3$.

Bezeichne zu einem Knoten x aus T die Tiefe von x mit $d(x, T)$. Dann ist die *innere Pfadlänge von* T, $P(T)$, definiert als

$$P(T) = \sum_{x \in T} d(x, T).$$

Wir wollen nun zeigen, dass $\mathbf{E}[P(T)] = O(n \log n)$ ist.

(b) Seien T_L und T_R der linke und der rechte Teilbaum von T. Zeigen Sie:

$$P(T) = P(T_L) + P(T_R) + |T| - 1.$$

(c) Sei $P(n)$ die erwartete innere Pfadlänge eines zufällig aufgebauten binären Suchbaums mit n Elementen. Folgern Sie aus der vorherigen Aufgabe, dass

$$P(n) = \frac{1}{n} \sum_{k=0}^{n-1} \left(P(k) + P(n-k-1) + n - 1 \right).$$

(d) Lösen Sie die Rekursionsgleichung aus (c) mit einer Methode Ihrer Wahl (machen Sie geeignete Annahmen über den Rekursionsanker).

(e) Folgern Sie, dass die durchschnittliche Tiefe eines Knoten in einem zufällig aufgebauten binären Suchbaum $O(\log n)$ ist.