Datenstrukturen SS 2019
Hans Simon
Daniel Pasler
Christoph Ries
Leonie Ryvkin

30. April 2019
Übungsblatt 4
Abgabe bis 7. Mai 2019, 12:00 Uhr auf IB 01

Aufgabe 4.1 (4 Punkte)
Betrachte das Alphabet $A = \{a, b, c, d, e, f\}$ mit der Häufigkeitsverteilung

$$F(a) = 0.29, F(b) = 0.19, F(c) = 0.22, F(d) = 0.07, F(e) = 0.21 \text{ und } F(f) = 0.02.$$

Führe die nicht-rekursive Variante des Huffman-Algorithmus bzgl. A und F aus. Gib dazu die Arrays A, N und den Heap H (als Liste wie im Beispiel - siehe Skriptteil) einmal zu Beginn des Algorithmus und dann nach *jedem* Schleifendurchlauf, sowie die Arrays LS, RS, LL *nur* nach Beendigung des Algorithmus an. Zeichne abschließend den resultierenden A-markierten Baum.

Aufgabe 4.2 (4 Punkte)
Es sei A ein (festes) Alphabet der Größe $n > 1$. Für eine Schlüsselwertverteilung f auf A sei γ_f der entstehende Huffman–Code. Gib eine Schlüsselwertverteilung F auf A an, sodass $ABL(\gamma_f)$ unter allen Wahlen f von Schlüsselwertverteilungen für $f = F$ maximal wird. D.h.

$$ABL(\gamma_F) = \max_f ABL(\gamma_f).$$

Zeige deine Behauptung.

Aufgabe 4.3 (4 Punkte)
Es sei T der durch folgende Array–Belegungen gegebene binäre Suchbaum:

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14
KEY	12	3	10	6	11	4	23	17	1	0	0	0	0	0
LS	4	9	0	2	0	0	8	0	0	0	0	0	0	0
RS	7	6	5	3	0	0	0	0	0	0	0	0	0	0

Die Elemente E entsprechen einfach den Schlüsselwerten selbst, die Wurzel ist $r = 1$. Wir verwenden folgende Konvention: Bei einem Aufruf von NEW wird immer die kleinste Adresse i zurückgeliefert, die möglich ist. Werte i mit KEY[i]=0 zeigen an, dass Adresse i zur Zeit nicht verwendet wird.

a) Füge in T nacheinander Elemente mit den Schlüsselwerten 8,5,26,14 (in dieser Reihenfolge) ein. Zeichne den entstehenden binären Suchbaum und gib die Belegungen der Arrays LS,RS und KEY an.

b) Lösche nun aus dem aus a) resultierenden Baum nacheinander die Elemente 10,6,12 (in dieser Reihenfolge) und zeichne den resultierenden Baum.

c) In einen anfangs leeren binären Suchbaum sollen Elemente mit aufsteigenden und paarweise verschiedenen Schlüsselwerten $a_1 < ... < a_k$ eingefügt werden. Wie kann man j wählen, sodass nach der ersten Einfügung des Elements mit Schlüsselwert a_j durch weiteres Einfügen immer noch ein Baum minimaler Höhe möglich ist? Begründe deine Behauptung.

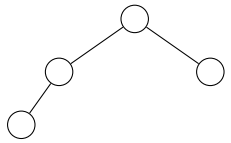
— Bitte wenden! —

Aufgabe 4.4 (4 Punkte)

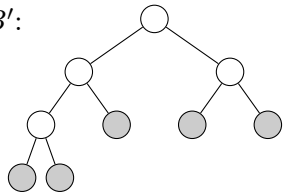
a) Sei B ein binärer Baum mit Knotenmenge V. Den erweiterten binären Baum zu B erhält man, indem man an jeden Knoten von B, der noch nicht zwei Kinder hat, einen bzw. zwei Knoten hinzufügt, sodass jeder Knoten aus V dann genau zwei Kinder besitzt. Diesen erweiterten Baum bezeichnen wir mit B' und die Menge seiner Knoten mit V'.

Beispiel:

B:



B':



Wir definieren nun die Länge des äußeren Pfades, $length_a(B)$, und die Länge des inneren Pfades, $length_i(B)$, von B.

$$\begin{aligned} length_a(B) &:= \sum_{v \in V' \setminus V} depth(v) \\ length_i(B) &:= \sum_{v \in V} depth(v) \end{aligned}$$

Zeige, dass zwischen der Länge des äußeren und der Länge des inneren Pfades folgender Zusammenhang besteht

$$length_a(B) = length_i(B) + 2|V|\,. \qquad (\star)$$

b) Sei $k \geq 2$ gegeben. Sei nun B_k ein Baum, bei dem jeder Knoten bis zu k Kinder besitzt. Der erweiterte Baum B'_k zu B_k ergebe sich nun, indem an jeden Knoten aus B_k der noch nicht k Kinder hat, Blätter hinzugefügt werden bis die Anzahl der Kinder genau k ist.
Stelle eine zu $(\star)$ analoge Gleichung für B_k auf und beweise sie.