Effiziente Algorithmen I SS 15

Prof. Dr. Meer, Dr. Gengler

Aufgabenblatt 4

Aufgabe 0
Bereiten Sie die Vorlesungen gewissenhaft nach, führen Sie dabei ein Zeitprotokoll. Sollten Sie weniger Zeit für die Nachbereitung verbrauchen als die Vorlesung dauerte, so wiederholen Sie diesen Schritt nochmal.

Aufgabe 1
Programmieren Sie den BFPRT-Algorithmus! Testen Sie Ihre Implementation u.a. mit einer Liste der länge 4711, welche überall den gleichen Eintrag hat, etwa 42. Gibt es Probleme?

Aufgabe 2
Wenden Sie mit Papier und Bleistift den BFPRT-Algorithmus auf die Liste (4,4,4,4,4,4,4,4,4,4,4,4,4,4,4) an. Gibt es Probleme?

Aufgabe 3
Bei Analyse von BFPRT-Algorithmus ergibt sich die Rekursionsgleichung:

$$T(n) \leq T(n/5) + T(7n/10 + 6) + O(n)$$

Versuchen Sie, das Master-Theorem anzuwenden. Etwa indem Sie $T(n/5)$ nach oben durch $T(7n/10)$ abschätzen und beide Argumente dann z.B. nach oben durch $8n/10$.

Aufgabe 4
Ändern Sie den BFPRT-Algorithmus wie im Folgenden beschrieben ab, und schätzen Sie dei Laufzeit der Variante ab:

1. Statt Fünfergruppen werden Dreiergruppen gebildet, von denne dann die Mediane gebildet werden, usw.
2. Statt Fünfergruppen werden Siebenergruppen gebildet, von denne dann die Mediane gebildet werden, usw.

Aufgabe 5
Für $a, b, c, d \in \{x \in \mathbb{R} \mid x > 1\}$ sei $f : \mathbb{N} \to \mathbb{R}_+$ definiert vermöge:

$$f(0) = d \quad \text{und} \quad f(n) = f(n/a) + f(n/b) + c{\cdot}n$$

1. Geben Sie eine (möglichst gute) Wachstumsklasse für f an.
2. Zeigen Sie, dass $f \in \mathcal{O}(\mathrm{id})$, falls $a{+}b < a{\cdot}b$ gilt.

Aufgabe 6
Wir betrachten die Fibonacci-Folge $f_0, f_1, f_2, \ldots$ definiert durch das rekursive Bildungsgesetz $f_n = f_{n-1} + f_{n-2}$ für $n \geq 2$ und den Anfangswerten $f_0 = 0$ und $f_1 = 1$. Sei $\Phi := \frac{1+\sqrt{5}}{2}$. Zeigen Sie:

$$\lim_{n\to\infty} \frac{f_{n+1}}{f_n} \quad = \quad \Phi$$

Hinweis: Nutzen Sie die Aufgabe 4 von Blatt 1 bzw. die Aufgaben 8 und 9 von Blatt 4.
Bemerkung: Φ bezeichnet den Goldenen Schnitt, siehe:
`http://mathworld.wolfram.com/GoldenRatio.html`

Aufgabe 7 (Fibonacci-Baum)
Wir betrachten AVL-Bäume:

1. Geben Sie zu jedem $h \in \mathbb{N}$ eine Folge von Insert- und Delete-Operationen an, die zu einem vollständigen binären Baum der Höhe h führt (d.h. bei dem alle Pfade von der Wurzel zu einem Blatt genau die Länge h haben).
2. Recherchieren Sie die Definition *Fibonacci-Baum*.
3. Bestimmen Sie die kleinste Länge eines Pfades von der Wurzel zu einem Blatt in einem Fibonacci-Baum der Höhe h.
4. Wie lang muss der kürzeste Pfad von der Wurzel zu einem Blatt bei einem AVL-Baum der Höhe h mindestens sein. Beweisen Sie Ihre Aussage. Geben Sie eine Folge von Insert- und Delete-Operationen an, so dass ein solcher kurzer Pfad entsteht.

Aufgabe 8 (Binomialer Baum)
Ein *binomialer Baum* B_k ist ein rekursiv definierter geordneter Baum. Der binomiale Baum B_0 besteht nur aus einem einzigen Knoten. Der binomiale Baum B_k besteht aus zwei binominialen Bäumen B_{k-1}, die miteinander verkettet sind: Die Wurzel des einen ist das linkeste Kind des andern ($k > 0$). Beweisen Sie:

1. B_k hat 2^k Knoten.
2. Die Höhe des Baumes B_k ist k.
3. In der Tiefe i hat der Baum B_k genau $\binom{k}{i}$ Knoten $(i = 0, 1, \ldots, k)$. (namensgebende Eigenschaft)
4. Die Wurzel von B_k hat den Grad k, der größer ist als der Grad jedes anderen Knoten in B_k.

Aufgabe 9 (Rot-Scharz-Baum)
Rot-Scharz-Bäume sind eine Variante von balabncierten binären Suchbäumen.

1. Recherchieren Sie die Definition *Rot-Schwarz-Baum*.
2. Zeigen Sie, dass in einem Rot-schwarz-Baum alle Pfade, welche von der Wurzel zu inem Blatt führen, höchsten doppelt so lang sind wie die Lände eines kürzesten Pfades von der Wurzel zu einem Blatt.
3. Zeigen Sie, dass die Höhe eines Rot-Schwarz-Baumes mit n Knoten in $\mathcal{O}(\log n)$ liegt.
4. Geben Sie an, wie die Insert- und Delete-Operationen implementiert werden können.