Algorithmen und Programmierung 3, WS 2019/2020 — 4. Übungsblatt

Abgabe bis Freitag, 16. November 2019, 12:00 Uhr, in die Fächer der Tutoren
Aufgabe 34a neu formuliert am 8. November, 12 Uhr.

---

26. Asymptotisches Wachstum, 0 Punkte

    Finden Sie möglichst einfache Ausdrücke der Form $\Theta(\cdot)$ für folgende Funktionen:

    (a) $3n^2 - 4n + 23 + 27\,n \cdot \lceil \log_2 n \rceil / 2$
    (b) $\max\{n\lceil \log_2 n\rceil, (\lceil \log_2 n\rceil)^4\}$
    (c) $2^{2n+\lceil \log_2 n\rceil}$
    (d) $1.8^{2^{\lceil \log_2 n\rceil}}$
    (e) $2^{\lfloor \log_4 n\rfloor}$
    (f) $2^{\lfloor \log_3 n\rfloor + \lceil \log_4 n\rceil}$

27. AVL-Bäume, 7 Punkte

    Fügen Sie nacheinander die Schlüssel 1, 20, 50, 40, 30, 35, 60, 70 in einen anfangs leeren AVL-Baum ein. Entfernen Sie dann die Schlüssel 35 und 1. Zeichnen Sie den AVL-Baum nach jeder Operation auf.

28. Rotationen und Doppelrotationen, Programmierung, 0 Punkte

    Schreiben Sie ein Programmstück für (a) eine Linksrotation, (b) eine Doppelrotation, die mit einer Linskrotation beginnt. Verwenden Sie dafür möglichst wenige Zuweisungen. Die Eingabe ist ein Zeiger auf die Wurzel des betroffenen Teilbaumes. Kommen Sie dabei ohne simultane Zuweisungen an mehrere Variablen gleichzeitig aus, wie es sie zum Beispiel in Python gibt.

29. Rotationen, 0 Punkte

    Beweisen Sie: Jeder Suchbaum für n Schlüssel kann durch einen Folge von Rotationen in jeden anderen Suchbaum für diese Schlüssel transformiert werden.

    Zusatzfrage (offen): Gilt diese Aussage auch für AVL-Bäume, wenn alle Zwischenergebnisse ebenfalls AVL-Bäume sein müssen?

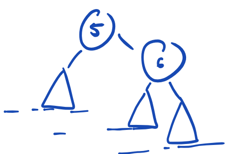
30. Erweiterte Suchanfragen für binäre Bäume, 0 Punkte

    Die Knoten `x` eines Suchbaumes enthalten neben den Kinderzeigern `x.links` und `x.rechts` auch den Schlüssel `x.s` und einen ganzzahligen Wert `x.w`. Wir wollen nun folgende Anfragen beantworten: Für zwei gegebene Schranken a und b soll die *Summe* alle Werte der Einträge bestimmt werden, deren Schlüssel s im Intervall $a < s \leq b$ liegt. Erwertern Sie die Baumknoten um geeignete Attribute, damit diese Anfragen in Laufzeit proportional zur Höhe des Baumes beantwortet werden können. Schreiben Sie dazu eine Funktion in Pseudocode, die als Eingabe a und b und einen Zeigers t auf die Wurzel des Baumes bekommt. Erläutern Sie auch, wie die neuen Attribute beim Erstellen oder beim Umbau des Baumes rekursiv berechnet werden.

31. Unterhaltungsaufgabe, 0 Punkte

    Herr B. C. Dom hat die Definition eines Suchbaums missverstanden als ein Baum, in dem der Schlüssel jedes Knotens größer ist als der Schlüssel im linken Kind und kleiner als der Schlüssel im rechten Kind (sofern diese Kinder vorhanden sind). Er hat nun eine Methode zum Patent angemeldet, mit der er eine ungeordnete Liste von n Schlüsseln in $O(n)$ Zeit in einen solchen „Suchbaum“ umwandeln kann.

    (a) Können Sie den Algorithmus von Herrn Dom erraten?
    (b) Wie viel würden Sie für die Lizenz zur Benützung dieses Patents zahlen?



32. Volle Binärbäume, 6 Punkte (zu unterscheiden von *vollständigen* Binärbäumen)

    (a) (0 Punkte) In einem *vollen* binären Baum hat jeder innere Knoten zwei Kinder. Man kann einen beliebigen Binärbaum als vollen Binärbaum darstellen, indem man für jedes fehlende Kind eines Knotens ein neues Blatt (einen *externen* Knoten) einsetzt. Alle ursprünglichen Knoten werden zu inneren Knoten, und die neuen Blätter repräsentieren die `null`/`Nil`-Zeiger im ursprünglichen Baum. Zeigen Sie, dass ein voller binärer Baum mit n Blättern $n-1$ innere Knoten hat.

    (b) (6 Punkte) Beweisen Sie, dass in einem vollen binären Baum mit n Blättern auf Tiefe $l_1, \ldots, l_n$ die folgende Gleichung gilt:

    $$\sum_{i=1}^{n} 2^{-l_i} = 1$$

    (c) (0 Punkte) Für jede Folge $l_1, \ldots, l_n$ ganzer Zahlen, die die obige Gleichung erfüllt, gibt es einen vollen binären Baum mit n Blättern auf Tiefe $l_1, \ldots, l_n$.

33. Skiplisten, 0 Punkte

    (a) Zeichnen Sie die Skipliste für sieben Schlüsselwerte $1, 2, \ldots, 7$ mit den jeweiligen Höhen $1, 2, 4, 3, 6, 1, 1$.

    (b) Illustrieren Sie die Suche nach den Schlüsselwerten 3.5 und 6.6.

    (c) Fügen Sie nacheinander die Schlüssel 1.4 mit Höhe 2 und 4.5 mit Höhe 3 ein.

    (d) Entfernen Sie die Schlüssel 2 und dann 4.

34. Analyse von Skiplisten, 7 Punkte

    L sei eine Skipliste mit n Einträgen. Zeigen Sie:

    (a) Die erwartete Gesamtanzahl von Elementen in allen Listen $L_1, L_2, \ldots, L_H$ von L zusammengenommen (ausgenommen die Listenköpfe) ist $2n$.

    (b) Die Wahrscheinlichkeit, dass L aus *mehr als k* Listen besteht, ist höchstens $n/2^k$. *Hinweis*: Für Ereignisse $A_1, \ldots, A_n$ gilt die *Vereinigungsschranke*: $\Pr[A_1 \vee \cdots \vee A_n] \leq \sum_{i=1}^{n} \Pr[A_i]$, wobei „Pr“ für Wahrscheinlichkeit steht.

35. Einfügen in Skiplisten, 0 Punkte

    Zeigen Sie, dass beim Einfügen im Mittel nur konstant viele Zeiger geändert werden. (Hinweis: Wie ist es beim Löschen?)

36. Skiplisten mit veränderter Dichte, 0 Punkte

    Wir legen fest, dass jedes Element von L_i mit Wahrscheinlicheit p in die Liste L_{i+1} aufgenommen wird, für ein festes p mit $0 < p < 1$ statt dem Standardfall $p = 1/2$.

    (a) Zeigen Sie, dass die erwartete Länge eines Suchpfades dann $\frac{1}{p} \log_{1/p} n + O(1)$ ist.

    (b) Welcher Wert von p minimiert diesen Ausdruck?

    (c) Was ist der Erwartungswert für die Anzahl der Zeiger?

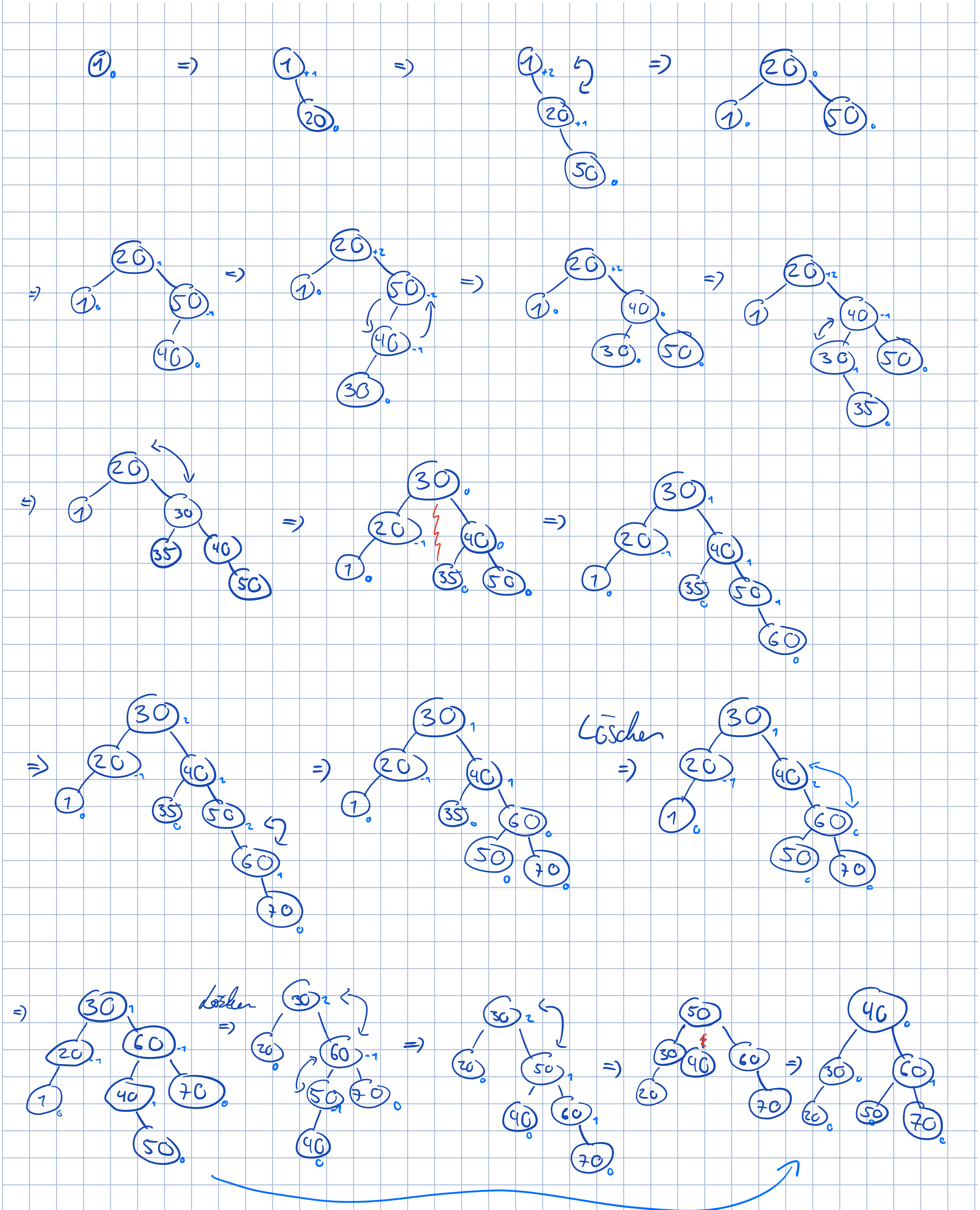
37. Skiplisten und Binärbäume

    (a) Zeigen Sie, wie man durch Entfernen einiger Kanten aus einer Skipliste eine Struktur herstellen kann, die wie ein Binärbaum aussieht und die ähnlich zu einem binären Suchbaum ist.

    (b) Skipisten und binäre Suchbäume verwenden ungefähr gleich viele Zeiger pro Knoten, nämlich 2. Erläutern Sie, in welchem Sinn diese Zeiger bei Skiplisten besser ausgenützt werden.

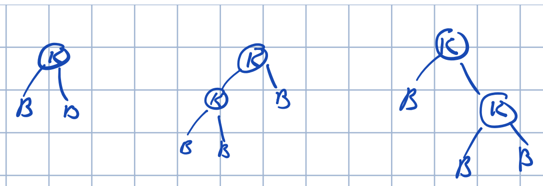
27. AVL-Bäume, 7 Punkte

Fügen Sie nacheinander die Schlüssel 1, 20, 50, 40, 30, 35, 60, 70 in einen anfangs leeren AVL-Baum ein. Entfernen Sie dann die Schlüssel 35 und 1. Zeichnen Sie den AVL-Baum nach jeder Operation auf.



(b) (6 Punkte) Beweisen Sie, dass in einem vollen binären Baum mit n Blättern auf Tiefe $l_1, \ldots, l_n$ die folgende Gleichung gilt:

$$\sum_{i=1}^{n} 2^{-l_i} = 1$$



$$\sum_{i=1}^{n} 2^{-l_i} = 1$$

I.A $n=2$ $\quad \sum_{i=1}^{2} 2^{-l_i} = 1$

$\iff 2^{-1} + 2^{-1} = 1$

$n+1$

$$\sum_{i=1}^{n+1} 2^{-l_i} = 1$$

$\ldots\ 2^{-l_1} + 2^{-l_2} + \ldots\ 2^{-l_{n+1}}$

$\frac{1}{2} + \frac{1}{2} \ldots + \frac{1}{n+1}$

$\frac{1}{4} + \frac{1}{4}$

$\vdots$

$\frac{1}{n+1}$

Rotation	BF Oberer Knoten	BF Unterer Knoten	
Rechts-Rotation	-2	-1	um OK
Links-Rotation	+2	+1	um OK
Rechts-Links-Rotation	+2	-1	R um UK, dann L um OK
Links-Rechts-Rotation	-2	+1	L um UK, dann R um OK

34. Analyse von Skiplisten, 7 Punkte

L sei eine Skipliste mit n Einträgen. Zeigen Sie:

(a) Die erwartete Gesamtanzahl von Elementen in allen Listen $L_1, L_2, \ldots, L_H$ von L zusammengenommen (ausgenommen die Listenköpfe) ist $2n$.

(b) Die Wahrscheinlichkeit, dass L aus *mehr als* k Listen besteht, ist höchstens $n/2^k$. *Hinweis*: Für Ereignisse $A_1, \ldots, A_n$ gilt die *Vereinigungsschranke*: $\Pr[A_1 \vee \cdots \vee A_n] \leq \sum_{i=1}^{n} \Pr[A_i]$, wobei „Pr“ für Wahrscheinlichkeit steht.

a)

k $\square$ Erwartungswert $\frac{1}{2^k} \cdot n = \frac{n}{2^k}$

$\frac{1}{2}$

$\boxed{1} \to \boxed{3} \ldots$

n lang

$Pr(h=1) = \frac{1}{2}$

$Pr(h=0) = 1$

$\sum_{i=0}^{\infty} 2^{-i} = 2$ Erwartungswert für h_i

$\Rightarrow$ Erwartungswert $\cdot$ Länge der Liste $= 2n$