TU Wien
Institut für Logic and Computation
Forschungsbereich Algorithms and Complexity

186.866 Algorithmen und Datenstrukturen VU

Übungsblatt 4

PDF erstellt am: 25. April 2024

Deadline für dieses Übungsblatt ist **Montag, 6.5.2024, 20:00 Uhr**. Damit Sie für diese Übung Aufgaben anerkannt bekommen können, gehen Sie folgendermaßen vor:

1. Öffnen Sie den TUWEL-Kurs der Lehrveranstaltung *186.866 Algorithmen und Datenstrukturen (VU 5.5)* und navigieren Sie zum Abschnitt *Übungsblätter*.

2. Teilen Sie uns mit, welche Aufgaben Sie gelöst haben **und** welche gelösten Aufgaben Sie gegebenenfalls in der Übungseinheit präsentieren können. Gehen Sie dabei folgendermaßen vor:

   - Laden Sie Ihre Lösungen in einem einzigen PDF-Dokument in TUWEL hoch.
     Link *Hochladen Lösungen Übungsblatt 4*
     Button *Abgabe hinzufügen* bzw. *Abgabe bearbeiten*
     PDF-Datei mit Lösungen hochladen und *Änderungen sichern*.
   - Kreuzen Sie an, welche Aufgaben Sie gegebenenfalls in der Übung präsentieren können. Die Lösungen der angekreuzten Aufgaben müssen im hochgeladenen PDF enthalten sein.
     Link *Ankreuzen Übungsblatt 4*
     Aufgaben entsprechend anhaken und *Änderungen speichern*.

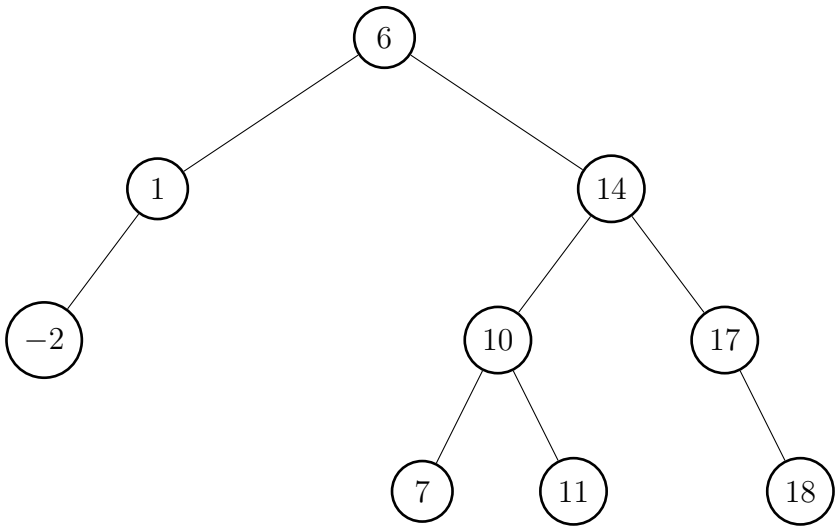
Bitte beachten Sie:

- Bis zur Deadline können Sie sowohl Ihr hochgeladenes PDF, als auch Ihre angekreuzten Aufgaben beliebig oft überschreiben. Sollte kurz vor der Deadline etwas schief gehen (Ausfall TUWEL, Internet, Scanner, etc.) und Sie die Endversion mit allen gelösten Aufgaben nicht mehr hochladen können, haben Sie zumindest Ihre Lösungen teilweise schon hochgeladen und angekreuzt. Nach der Deadline ist keine Veränderung mehr möglich. Die Deadline ist strikt – es werden ausnahmslos keine Nachabgabeversuche (z.B. per E-Mail) akzeptiert.

- Sie können Ihre Lösungen entweder direkt in einem Textverarbeitungsprogramm erstellen und hochladen, oder aber auch gut leserliche Scans bzw. Fotos von handschriftlichen Ausarbeitungen hochladen (beachten Sie die maximale Dateigröße).

- Beachten Sie die Richtlinien für das An- und Aberkennen von Aufgaben. Details dazu finden Sie in den Folien der Vorbesprechung.

Aufgabe 1. Gegeben ist folgender **AVL-Baum**:



(a) Fügen Sie zuerst den Schlüssel -9 und dann 13 in den oben gegebenen AVL-Baum ein. Falls notwendig, so rebalancieren Sie den Baum nach jedem Einfügen mit geeigneten Rotationsoperationen (siehe Foliensatz „Suchbäume“), um wieder einen gültigen AVL-Baum zu erhalten. Markieren Sie dabei den unbalancierten Knoten mit maximaler Tiefe.

(b) Löschen Sie aus dem ursprünglichen in Aufgabe 1 gegebenen **AVL-Baum** die Schlüssel -2 und 6 in dieser Reihenfolge. Falls notwendig, so rebalancieren Sie den Baum nach jedem Löschvorgang mit geeigneten Rotationsoperationen (siehe Foliensatz „Suchbäume“), um wieder einen gültigen AVL-Baum zu erhalten. Markieren Sie dabei den unbalancierten Knoten mit maximaler Tiefe.

Aufgabe 2. Gegeben sei die folgende Folge von Elementen:

$$\langle 10, 30, 60, 80, 110, 140, 200, 400 \rangle$$

(a) Fügen Sie die Elemente in dieser Reihenfolge in einen anfangs leeren B-Baum der Ordnung 3 ein. Zeichnen Sie den B-Baum jeweils vor und nach jeder Reorganisationsmaßnahme und geben Sie den endgültigen B-Baum an.

(b) Fügen Sie die Folge auch in einen natürlichen binären Suchbaum ein und vergleichen Sie das Resultat mit dem B-Baum aus Unteraufgabe (a).

(c) Geben Sie den B-Baum an, der durch Löschen der Schlüssel 110 und 80 (in dieser Reihenfolge) aus dem B-Baum von Unteraufgabe (a) entsteht.

Aufgabe 3. Geben Sie einen Algorithmus in detailliertem Pseudocode an, der folgendes Problem löst:

Die Eingabe ist ein Array A. Falls A einer gültigen **postorder** Traversierung eines **binären Suchbaums** entspricht, soll Ihr Algorithmus einen solchen binären Suchbaum erstellen und dessen Wurzelknoten zurückgeben. Andernfalls soll der Algorithmus als Rückgabewert *false* zurückgeben. Sie können annehmen, dass alle Elemente in A paarweise unterschiedliche natürliche Zahlen sind. Ihnen stehen dafür folgende Operationen zur Verfügung:

- NEWNODE(x): diese Methode erstellt einen neuen Knoten mit Wert x und leerem linken und rechten Kind und gibt diesen neuen Knoten zurück.
- v.left $\leftarrow v_\ell$: dies weist dem linken Kind von dem Knoten v den Knoten v_ℓ zu. Das heißt, danach ist v_ℓ das linke Kind von v.
- v.right $\leftarrow v_r$: dies weist dem rechten Kind von dem Knoten v den Knoten v_r zu. Das heißt, danach ist v_r das rechte Kind von v.

Hinweis 1: Es gibt maximal einen binären Suchbaum, dessen postorder Traversierung A entspricht. Wieso?
Hinweis 2: Versuchen Sie den Algorithmus als rekursive Funktion zu beschreiben.

Aufgabe 4. Gegeben ist folgende Hashtabelle der Größe $m = 11$:

Position	0	1	2	3	4	5	6	7	8	9	10
Schlüssel	44			3			17	7	30	20	

Führen Sie folgende Schritte in der vorgegebenen Reihenfolge aus:

1. Füge 13 ein
2. Füge 6 ein
3. Lösche 7
4. Suche 6
5. Füge 28 ein

Tun Sie dies für **jede** der folgenden Varianten zur Kollisionsbehandlung. Beschreiben Sie die einzelnen Schritte und stellen Sie die finale Belegung dar.

(a) Verkettung der Überläufer mit $h(k) = k \bmod m$.

(b) Lineares Sondieren mit $h(k, i) = (h'(k) + i) \bmod m$ und $h'(k) = k \bmod m$.

(c) Quadratisches Sondieren mit $h(k, i) = (h'(k) + \frac{1}{2}i + \frac{1}{2}i^2) \bmod m$ und $h'(k) = k \bmod m$.

Für das lineare und quadratische Sondieren gilt $i = 0, 1, \ldots, m - 1$. Nehmen Sie beim linearen und quadratischen Sondieren an, dass zu Beginn alle leeren Felder in der Hashtabelle mit dem Flag „frei“ markiert sind.

Aufgabe 5. Gegeben ist folgende Hashtabelle der Größe $m = 7$.

Position	0	1	2	3	4	5	6
Schlüssel	42		2	3		12	

Wir verwenden Double Hashing mit $h_1(k) = k \bmod m$ und $h_2(k) = (k \bmod 5) + 1$.

(a) Fügen Sie 66 in die obige Tabelle ein. Geben Sie alle Zwischenschritte an.

(b) Fügen Sie erneut 66 in die obige (ursprüngliche) Tabelle ein. Verwenden Sie diesmal dazu das verbesserte Einfügen nach Brent. Geben Sie alle Zwischenschritte an.

(c) Bestimmen Sie die durchschnittliche Anzahl an Schritten für eine erfolgreiche Suche in der Tabelle die Sie in Unteraufgabe (a) nach dem Einfügen erhalten. Gehen Sie analog für die Tabelle, die Sie aus Unteraufgabe (b) erhalten, vor.
