Algorithmen und Programmierung 3, WS 2019/2020 — 5. Übungsblatt

Abgabe bis Freitag, 22. November 2019, 12:00 Uhr, in die Fächer der Tutoren

---

38. Mittlere innere Weglänge, 0 Punkte

    (a) Fügen Sie die Elemente $1, 2, 3, 4, 5, 6, 7$ in dieser Reihenfolge einen leeren binären Suchbaum ein und bestimmen Sie die *mittlere innere Weglänge* des entstehenden Baumes; sie ist um 1 kleiner als die erwartete Anzahl von Knoten, die beim Finden eines zufällig ausgewählten Knotens im Baum besucht werden. Wir nehmen dabei an, dass jeder Knoten mit gleicher Wahrscheinlichkeit gesucht wird.

    (b) Wie ändert sich die Antwort, wenn man einen AVL-Baum verwendet?

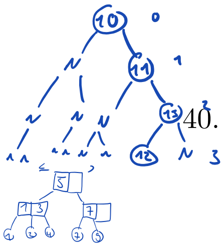
    (c) Lösen Sie Aufgabe (a) und (b) auch für die Einfügereihenfolge $4, 2, 3, 6, 7, 1, 5$.

39. Mittlere Weglänge, 6 Punkte (Programmieraufgabe, eine ehemalige Klausuraufgabe)

    Schreiben Sie ein Programm in JAVA, HASKELL oder PYTHON, das für einen gegebenen binären Suchbaum die mittlere innere Weglänge berechnet.

    Ihr Programm muss übersetzbar und lauffähig sein, aber *nicht* vollständig in dem Sinn, dass Sie zum Beispiel das Suchen und alle notwendigen und nützlichen Methoden für eine Klasse `Baum` programmieren müssen. Beschränken Sie sich auf die gestellte Aufgabe. Man muss das Programm testen können. Sie können Aufgabe 16 verwenden, um Testeingaben zu erstellen. Dokumentieren Sie genau, welche Form der Eingabe Ihr Testprogramm erwartet. Wenn das Programm als Hauptprogramm gestartet wird, muss es mindestens drei Testbeispiele automatisch verarbeiten.

$\frac{8}{5}$ = Mittlerweglänge

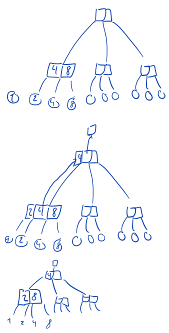


B+ Bäume

40. 2-3-Bäume, 7 Punkte

    Konstruieren Sie für jede Höhe $H \geq 1$ einen 2-3-Baum mit Höhe H und folgender Eigenschaft: Man kann ein neues Element x einfügen, sodass man den gesamten Weg zur Wurzel durchlaufen muss und sich die Höhe um 1 erhöht. Beim anschließenden Entfernen dieses Elementes x wird der ursprüngliche Zustand wiederhergestellt.

    (Bei 2-4-Bäumen kann das nicht passieren.)



41. Speicherersparnis bei a-b-Bäumen, 0 Punkte

    Wir wollen eine Variante von a-b-Bäumen konstruieren, bei der weniger Speicher verschwendet wird. Es sei $b = 100$. Wie groß kann man a wählen, wenn man

    - beim Entfernen bis zu *zwei* benachbarte Geschwisterknoten zum Ausborgen in Betracht zieht, und
    - beim Einfügen einen Geschwisterknoten zu Hilfe nimmt, falls dieser den Überlauf aufnehmen kann?

42. Rot-Schwarz-Bäume, 0 Punkte

    Rot-Schwarz-Bäume sind binäre Suchbäume mit folgenden Eigenschaften:[1]

    (a) Jeder innere Knoten hat zwei Kinder.

    (b) Jeder Knoten ist entweder als *rot* oder als *schwarz* gekennzeichnet.

    (c) Die Wurzel und alle Blätter sind schwarz.

---

[1] Es gibt verschiedene Versionen von Rot-Schwarz-Bäumen. In der Regel enthalten die inneren Knoten auch Werte und nicht nur Schlüssel (im Gegensatz zu den a-b-Bäumen, wie sie in der Vorlesung besprochen wurden). Rot-Schwarz-Bäume sind in der Klasse `TreeMap` im JAVA-Paket `java.util` implementiert.

(d) Die Kinder eines roten Knotens sind schwarz.

(e) Alle Wege von der Wurzel zu den Blättern enthalten die gleiche Anzahl von schwarzen Knoten.

Die um eins verminderte Anzahl der schwarzen Knoten in Eigenschaft (e) nennt man die *schwarze Höhe*.

1. Zeichnen Sie Rot-Schwarz-Bäume mit 5, 7 und 12 Blättern. h' $2h'$
2. Welche Höhe kann ein Baum mit schwarzer Höhe h' mindestens und höchstens haben?
3. $\sum_{i=1}^{h} 2^{i\cdot 2-1}$ Wie viele rote Knoten kann ein Baum mit schwarzer Höhe h' mindestens und höchstens haben? 0 $\sum_{i=0}^{h'} 2^i = 2^{h'+1}-1$
4. $\sum_{i=0}^{h} 2^{i\cdot 2} = \sum_{i=0}^{h} 4^i$ Wie viele schwarze Knoten kann ein Baum mit schwarzer Höhe h' mindestens und höchstens haben?
5. Zeigen Sie, dass man aus jedem Rot-Schwarz-Baum einen 2-4-Baum machen kann, und umgekehrt.

Begründen Sie Ihre Antworten.
(Fragen 2–4 sind abgewandelte ehemalige Klausuraufgaben.)

43. Integrieren, 7 Punkte

Das Integral einer stückweise konstanten Funktion $g\colon [a,b) \to \mathbb{R}$ ist eine stückweise lineare stetige Funktion $f\colon [a,b] \to \mathbb{R}$, wie in Aufgabe 22 vom 3. Zettel definiert:

$f(x) = 3x^2 + 2x + 3$

$$f(x) = \int_{u=a}^{x} g(u)\, du$$

treppen-funktion
Ottoforster analysis 3 Kapitel 18 riemansches integral

(a) (Vorbereitung, 0 Punkte) Denken Sie sich eine stückweise konstante Funktion g mit mindestens drei Stücken aus, und berechnen Sie dann f. Für welche Eingabe g ergibt sich die Beispielfunktion f, die bei Aufgabe 22 abgebildet ist?

(b) (Programmieraufgabe, 7 Punkte)
Schreiben Sie ein Programm in Python, Haskell oder Java, um eine Darstellung von f aus g zu berechnen. *Erläutern Sie, welche Darstellung von f Sie gewählt haben.* Programmieren Sie auch die Auswertung einer stückweise linearen Funktion $f(x)$ an der Stelle x.

(c) (Zusatzaufgabe, 5 Zusatzpunkte) Wenn $f\colon [a,b] \to \mathbb{R}$ eine stückweise lineare stetige Funktion ist, dann ist die rechtsseitige Ableitung $f^+(x) = \lim_{t\to 0+} \frac{f(x+t)-f(x)}{t}$ eine stückweise konstante Funktion $f^+\colon [a,b) \to \mathbb{R}$. Schreiben Sie ein Programm zur Berechnung von f^+ aus f.

Dokumentieren Sie genau, in welcher Form Ihre Programme die Eingabe erwarten. Wenn das Programm als Hauptprogramm gestartet wird, muss es mindestens drei Testbeispiele automatisch verarbeiten.

44. Suffixbaum, 0 Punkte

Zeichen Sie den Suffixbaum für *abaababaabaababaababaabaababaabaab* (das sogenannte *Fibonacci-Wort*). Wie geht die Folge weiter?

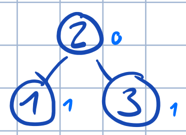
45. ShinSort, 0 Punkte

Versuchen Sie, das Sortierverfahren *Shin Sort* und die *Shin-Bäume*[2] des „weltgrößten Informatikers“ Dr. Dong-Keun Shin zu verstehen. Können Sie das mit einer Ihnen bekannten Methode oder Datenstruktur in Beziehung setzen?

[2] `http://www.dkshin.com/`

39 **Test Aufgaben**

1)

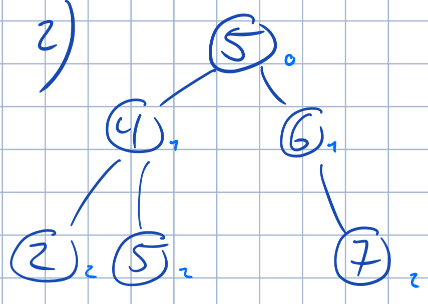


Anzahl der Elem

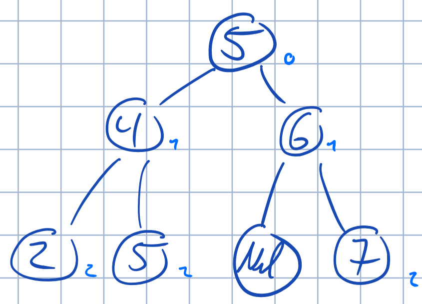
$xs = [3, 2, 1, 3]$

$\Rightarrow \frac{2}{3}$

2)



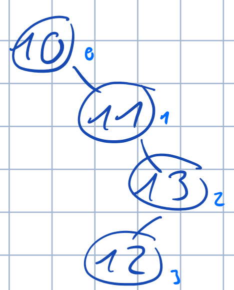
$\Leftrightarrow$



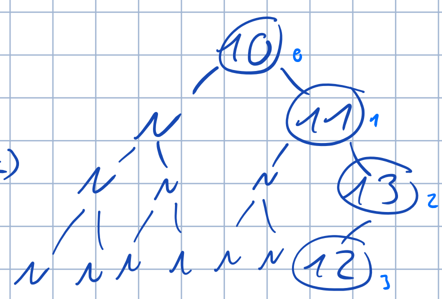
$ys = [6, 5, 4, 6, 2, 5, None, 7]$

$\Rightarrow \frac{8}{6}$

3)



$\Leftrightarrow$

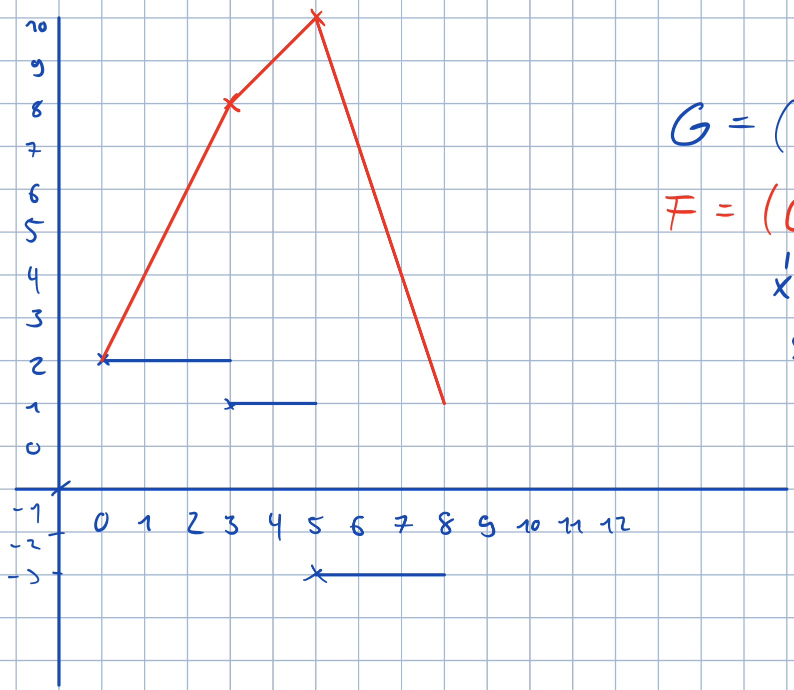


$\Rightarrow \frac{6}{4}$

$zx = [4, 10, N, 11, N, N, N, 13, N, N, N, N, N, N, 12]$

43

f definiert aus einer Liste aus "gedachten dreier Tupeln"
in der Reihenfolge x, Steigung, y ...

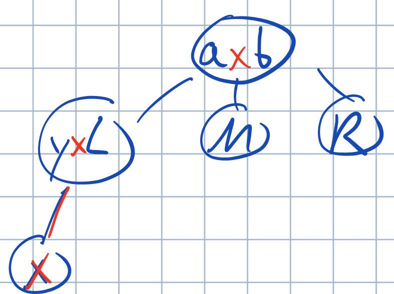


$G = (0, 2, 3, 1, 5, -3)$

$F = (0, 2, 2, 3, 1, 8, 5, -3, 10)$

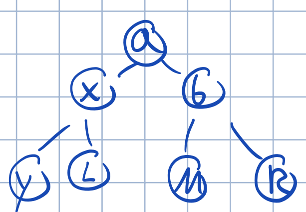
x_1 | Steigung | y_1 | x_2 | Steigung | y_2

40



f

⇒ x einfügen



Rot schwarz Baum