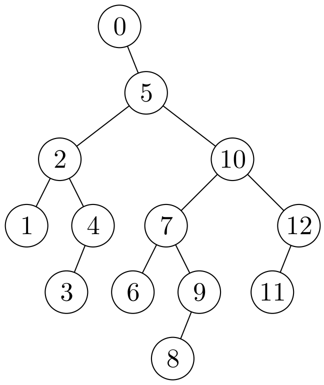
Datenstrukturen und Algorithmen, SS 2024
Prof. Dr. P. Rossmanith
D. Cloerkes, M. Gehnen, D. Mock, N. Speetzen

Abgabe: 26.04.2024

Übungsblatt mit Lösungen 02

Aufgabe T4

Gegeben ist folgender Binärbaum:



Geben Sie jeweils den entstandenen Baum an, nachdem folgende Operationen ausgeführt wurden. Führen Sie die Operationen jeweils auf dem resultierenden Baum der vorherigen Teilaufgabe aus.

a) Löschen Sie die 0.

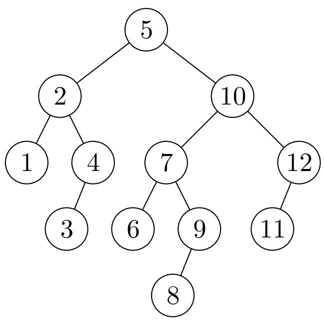
b) Löschen Sie die 10.

c) Fügen Sie die 10 ein.

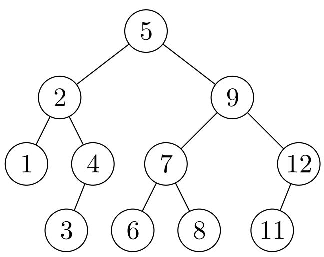
d) Löschen Sie die 5.

Lösungsvorschlag

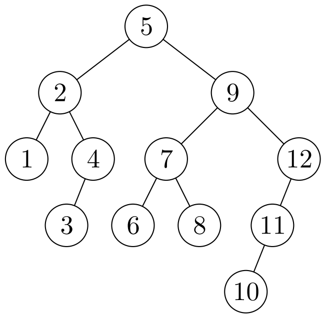
Löschen Sie die 0:



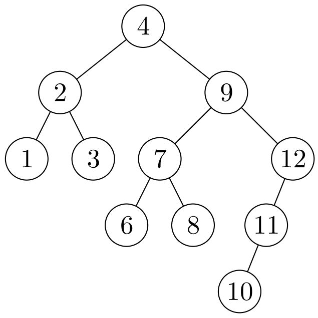
Löschen Sie die 10:



Fügen Sie die 10 ein:



Löschen Sie die 5:



Aufgabe T5

Wir fangen mit einem leeren AVL-Baum an. Führen Sie die Schritte a) bis e) nacheinander aus und zeichnen Sie den Binärbaum nach jedem Schritt.

a) Fügen Sie die Zahlen 1,...,5 in dieser Reihenfolge ein.

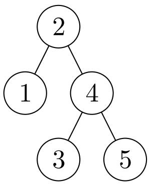
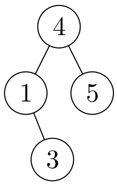
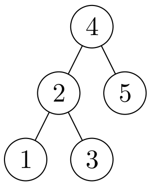
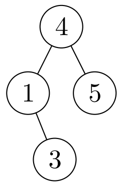
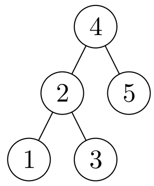
b) Löschen Sie die 2.

c) Fügen Sie die 2 ein.

d) Löschen Sie die 2.

e) Fügen Sie die 2 ein.

Lösungsvorschlag

a) 
b) 
c) 
d) 
e) 

Aufgabe T6

Konstruieren Sie einen optimalen Suchbaum für die Wörter RWTH, ETH, MIT, TUM und KIT (bezüglich der lexikographischen Ordnung). Auf diese werde mit den Wahrscheinlichkeiten 0.2, 0.25, 0.35, 0.15 und 0.05 zugegriffen.
Erstellen Sie die Tabellen für $w_{i,j}$ und $e_{i,j}$.

Lösungsvorschlag

Die Tabellen sehen wie folgt aus:

$w_{i,j}$	ETH	KIT	MIT	RWTH	TUM
ETH	0.25	0.30	0.65	0.85	1.00
KIT	0.0	0.05	0.40	0.60	0.75
MIT	0.0	0.0	0.35	0.55	0.70
RWTH	0.0	0.0	0.0	0.20	0.35
TUM	0.0	0.0	0.0	0.0	0.15

$e_{i,j}$	ETH	KIT	MIT	RWTH	TUM
ETH	0.25(ETH)	0.35(ETH)	1.00(MIT)	1.40(MIT)	1.85(MIT)
KIT	0.0	0.05(KIT)	0.45(MIT)	0.85(MIT)	1.30(MIT)
MIT	0.0	0.0	0.35(MIT)	0.75(MIT)	1.20(MIT)
RWTH	0.0	0.0	0.0	0.20(RWTH)	0.50(RWTH)
TUM	0.0	0.0	0.0	0.0	0.15(TUM)

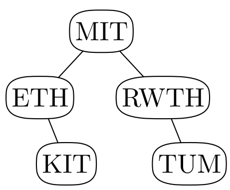
Für $i \leq j$ berechnen wir die einzelnen Werte mit der Formel

$$e_{i,j} = \min_{i \leq r \leq j} (e_{i,r-1} + e_{r+1,j}) + w_{i,j}.$$

In den Klammern hinter den Werten steht, welches r jeweils gewählt wurde, um diesen Wert zu erreichen. Beispielsweise gilt für $e_{\text{ETH,MIT}}$:

$$\begin{aligned}
e_{\text{ETH,MIT}} &= \min_{\text{ETH} \leq r \leq \text{MIT}} (e_{\text{ETH},r-1} + e_{r+1,\text{MIT}}) + w_{\text{ETH,MIT}} \\
&= \min(e_{\text{ETH,ETH-1}} + e_{\text{KIT,MIT}}, e_{\text{ETH,ETH}} + e_{\text{MIT,MIT}}, e_{\text{ETH,KIT}} + e_{\text{RWTH,MIT}}) + 0.65 \\
&= \min(0 + 0.45, 0.25 + 0.35, 0.35 + 0) + 0.65 \\
&= 0.35 + 0.65
\end{aligned}$$

Hier wurde $r = \text{MIT}$ gewählt, um den minimalen Wert 1.00 zu erhalten, also muss die Wurzel des Teilbaums mit allen Schlüsseln zwischen ETH und MIT (also den Schlüsseln ETH, KIT und MIT) der Knoten mit dem Schlüssel MIT sein. Da wir für $e_{\text{ETH,KIT}}$ $r = \text{ETH}$ gewählt haben, ist ETH die Wurzel des Teilbaums mit den Schlüsseln ETH und KIT. Zusammen mit der Bedingung, dass der Baum die Suchbaumeigenschaft erfüllt, ergibt sich eine eindeutige Anordnung des Teilbaums mit den Schlüsseln ETH, KIT und MIT. Ebenso verfahren wir mit den Teilbäumen, die die weiteren Schlüssel RWTH und TUM enthalten. So entsteht folgender optimaler Suchbaum.



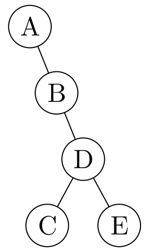
Aufgabe H5 (10 Punkte)
Gegeben sind zwei binäre Suchbäume, welche genau die gleichen Schlüssel enthalten. Kann immer der eine Baum in den anderen überführt werden, indem man nur Rotationen durchführt? Wenn ja, dann beschreiben Sie einen Algorithmus dafür. Wenn nein, dann finden Sie ein Gegenbeispiel.

Lösungsvorschlag
Wir beweisen per Induktion über die Struktur eines Binärbaums, dass ein beliebiger Binärbaum nur mit Rotationen in eine verkettete Liste von rechten (bzw. in eine verkettete Liste von linken) Kindern überführt werden kann. Wenn wir dann den Baum t_1 in einen Baum t_2 mit

gleichen Schlüsseln überführen wollen, können wir t_1 zunächst in eine Liste von rechten Kindern überführen. Da t_2 in die gleiche Liste überführt werden kann, kann diese Liste auch (mit inversen Rotationen) in t_2 überführt werden.

Induktionsanfang: Wenn wir nur einen Knoten haben, sind keine Rotationen nötig, da die Bäume bereits identisch sind.
Induktionsschritt: Wir beweisen die Aussage für einen Baum mit Wurzel w, linkem Unterbaum t_l mit n_l Knoten und rechtem Unterbaum t_r mit n_r Knoten unter der Voraussetzung, dass die Aussage für t_l und t_r gilt. Wir überführen t_l in eine verkettete Liste von linken Kindern und t_r in eine verkettete Liste von rechten Kindern (Induktionsvoraussetzung). Nun können wir den Baum in eine verkettete Liste von rechten Kindern überführen, indem wir n_l-mal um die Wurzel nach rechts rotieren. Analog können wir den Baum in eine verkettete Liste von linken Kindern überführen, indem wir n_r-mal um die Wurzel nach links rotieren.

Aufgabe H6 (10 Punkte)
Gegeben sei der folgende Suchbaum, in dem die Elemente A, B, C, D, E lexikographisch geordnet sind. Beweisen oder widerlegen Sie:
Der Baum kann ein optimaler Suchbaum mit 2.1 erwarteten Vergleichen sein.



Lösungsvorschlag
Die Aussage ist falsch.
Beweis:
1) Element A wird mit einer Wahrscheinlichkeit von mindestens 0.5 gesucht.
Würde A mit einer geringeren Wahrscheinlichkeit k_A gesucht werden, hätte ein Suchbaum mit B als Wurzel und A als linkes Kind von B einen geringeren Erwartungswert der Anzahl der Vergleiche: Das Suchen nach A wäre jeweils um 1 teurer, dafür das Suchen nach jedem anderen Element um 1 günstiger. Da $k_A < \frac{1}{2}$ wäre der Baum mit B als Wurzel günstiger.
2) In dem Teilbaum B, C, D, E mit B als Wurzel wird B mit einer Wahrscheinlichkeit w_B von mindestens $\frac{2}{5}$ gesucht.
Ist w_B kleiner als $2w_C$, so wäre der Teilbaum mit C als Wurzel, B als linkem Kind von C und der Teilbaum mit D, E rechts von C günstiger: eine Suche nach B ist um 1 teurer, eine Suche nach C um 2 günstiger. Aus $w_B < 2w_C$ folgt dann, dass der Teilbaum mit C als Wurzel günstiger ist. Die Kosten von D und E bleiben gleich.
Ist w_B kleiner als $w_D + w_E$, so wäre der folgende Teilbaum mit D als Wurzel günstiger: E sei das rechte Kind von D, B das linke. C sei das rechte Kind von B. Dann ist die Suche nach B um eins teurer, und die Suche nach D und E jeweils um 1 günstiger. Die Kosten für C ändern sich nicht. Ist $w_B < w_D + w_E$, so wäre dieser Teilbaum günstiger.
Aus den Bedingungen $w_B \geq 2w_C$, $w_B \geq w_D + w_E$ und $w_B + W_C + w_D + w_E = 1$ folgt, dass w_B mindestens $\frac{2}{5}$ sein muss (für $w_B = \frac{2}{5}, w_C = \frac{1}{5}, w_D + w_E = \frac{2}{5}$ sind beide Ungleichungen mit Gleichheit erfüllt).
3) In dem Teilbaum C, D, E mit D als Wurzel muss D mit einer Wahrscheinlichkeit w_D von mindestens $\frac{1}{5}$ gesucht werden.
Ist w_D kleiner als $\frac{1}{5}$ so ist der Teilbaum mit Höhe 3, in dem alle Elemente nach Zugriffswahrscheinlichkeit sortiert sind (das am öftesten gesuchte Element als Wurzel, das am seltensten gesuchte als Kind) günstiger: Das am öftesten Gesuchte Element wird nach dem Schubfachprinzip mit einer Wahrscheinlichkeit von mindestens $\frac{2}{5}$ gesucht, das am seltensten Gesuchte Element mit einer Wahrscheinlichkeit von maximal $\frac{1}{5}$ (da schon $w_D < \frac{1}{5}$). Der Erwartungswert der Zugriffe sind also $1 + \frac{3}{5} + w_D < \frac{9}{5}$.

Da der Baum mit D als Wurzel im Erwartungswert $1 + (1 - w_D) > \frac{9}{5}$ Zugriffe benötigt, kann er kein optimaler Teilbaum des gesamten Suchbaums sein.

Zusammen:

Die Elemente der vierten Ebene (C, E) werden (zusammen) maximal mit einer Wahrscheinlichkeit von $(1 - w_A)(1 - w_B)(1 - w_D) \leq 0.5\frac{3}{5}\frac{4}{5}$ gesucht.

Die Element der mindestens dritten Ebene (C, D, E) werden maximal mit einer Wahrscheinlichkeit von $(1 - w_A)(1 - w_B) \leq 0.5\frac{3}{5}$ gesucht.

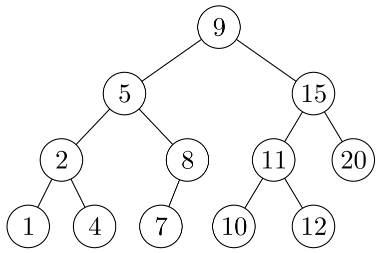
Die Elemente der mindestens zweiten Ebene (B, C, D, E) werden maximal mit einer Wahrscheinlichkeit von $(1 - w_A) \leq 0.5$ gesucht.

Die Elemente der mindestens ersten Ebene (A, B, C, D, E) werden mit einer Wahrscheinlichkeit von 1 gesucht.

Im Erwartungswert kann es also maximal $0.5\frac{4}{5}\frac{3}{5} + 0.5\frac{3}{5} + 0.5 + 1 = 2.04$ Vergleiche geben. Da dies eine obere Schranke ist, kann es keine Suchwahrscheinlichkeiten $w_A, ..., w_E$ geben, sodass der gegebene Baum ein optimaler Suchbaum ist.

Aufgabe H7 (10 Punkte)

Gegeben ist folgender AVL-Baum:



Wenden Sie im folgenden Operationen immer auf den entstandenen Baum der vorherigen Teilaufgabe an.

a) In welcher Reihenfolge könnten die Schlüssel eingefügt worden sein, so dass gerade dieser AVL-Baum entsteht?

b) Wie sieht der Baum aus, wenn wir die 13 einfügen?

c) Was erhalten wir, wenn 7 und 8 gelöscht werden?

d) Jetzt wird die 9 gelöscht. Wie sieht der Baum danach aus?

e) Zuletzt fügen wir 14 ein. Was erhalten wir dadurch?

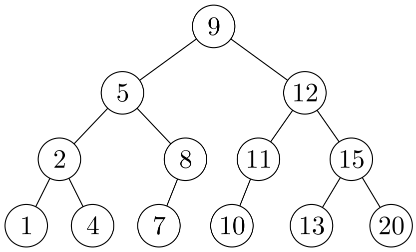
Lösungsvorschlag

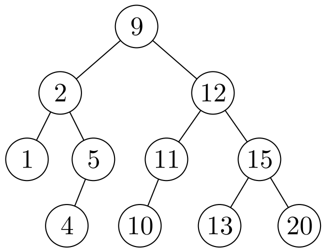
a) Am einfachsten ist es einzusehen, dass die Schlüssel „schichtenweise“ von oben nach unten eingefügt worden sein können. Dabei finden gar keine Rotationen statt, da der Baum immer balanciert bleibt. Diese Reihenfolge ist, wenn wir von links nach rechts gehen:

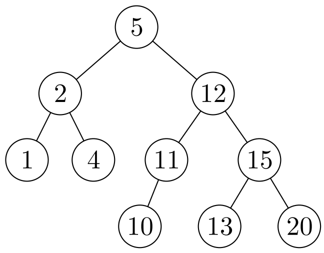
$$9, 5, 15, 2, 8, 11, 20, 1, 4, 7, 10, 12$$

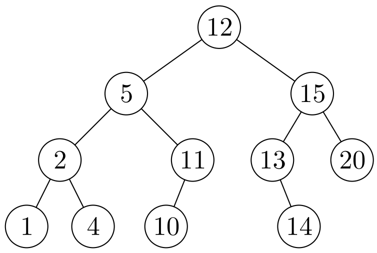
Es gibt natürlich viele andere Möglichkeiten. Allein wenn wir schichtenweise vorgehen, gibt es $2! \cdot 4! \cdot 5! = 5760$ Möglichkeiten.

Insgesamt gibt es aber noch viel mehr Reihenfolgen, nämlich genau 7,076,160.

b) 
9
5
12
2
8
11
15
1
4
7
10
13
20

c) 
9
2
12
1
5
11
15
4
10
13
20

d) 
5
2
12
1
4
11
15
10
13
20

e) 
12
5
15
2
11
13
20
1
4
10
14