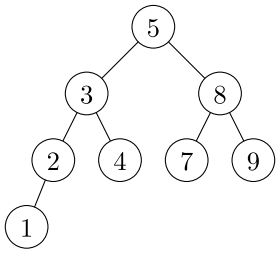
Datenstrukturen und Algorithmen, SS 2019
Prof. Dr. P. Rossmanith
S. Dollase, J. Dreier, H. Lotze

Datum: 18.04.2019

Übungsblatt mit Lösungen 02

Aufgabe T5

Gegeben sei dieser binäre Suchbaum:



a) In welcher Reihenfolge könnten die Schlüssel in einen anfangs leeren Suchbaum eingefügt worden sein?

b) Gibt es mehr als sieben verschiedene Einfügereihenfolgen, wenn jeder Schlüssel genau einmal eingefügt wurde?

c) Wie sieht der Baum aus, wenn 6 eingefügt wird?

d) Wie sieht er aus, wenn jetzt die 5 gelöscht wird?

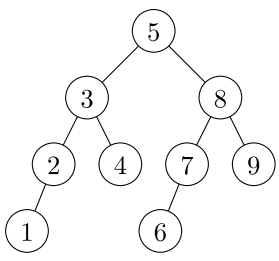
Lösungsvorschlag

a) Z.B.: $5, 3, 8, 2, 4, 7, 9, 1$

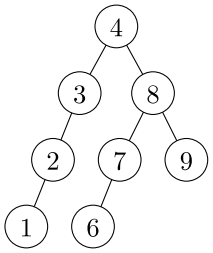
b) Es gibt mehr, nämlich genau 210. Hier sind sie:

53214879 53214897 53218479 53218497 53218749 53218794 53218947 53218974 53241879 53241897 53248179 53248197 53248719 53248791
53248917 53248971 53281479 53281497 53281749 53281794 53281947 53281974 53284179 53284197 53284719 53284791 53284917 53284971
53287149 53287194 53287419 53287491 53287914 53287941 53289147 53289174 53289417 53289471 53289714 53289741 53421879 53421897
53428179 53428197 53428719 53428791 53428917 53428971 53482179 53482197 53482719 53482791 53482917 53482971 53487219 53487291
53487921 53489217 53489271 53489721 53821479 53821497 53821749 53821794 53821947 53821974 53824179 53824197 53824719 53824791
53824917 53824971 53827149 53827194 53827419 53827491 53827914 53827941 53829147 53829174 53829417 53829471 53829714 53829741
53842179 53842197 53842719 53842791 53842917 53842971 53847219 53847291 53847921 53849217 53849271 53849721 53872149 53872194
53872419 53872491 53872914 53872941 53874219 53874291 53874921 53879214 53879241 53879421 53892147 53892174 53892417 53892471
53892714 53892741 53894217 53894271 53894721 53897214 53897241 53897421 58321479 58321497 58321749 58321794 58321947 58321974
58324179 58324197 58324719 58324791 58324917 58324971 58327149 58327194 58327419 58327491 58327914 58327941 58329147 58329174
58329417 58329471 58329714 58329741 58342179 58342197 58342719 58342791 58342917 58342971 58347219 58347291 58347921 58349217
58349271 58349721 58372149 58372194 58372419 58372491 58372914 58372941 58374219 58374291 58374921 58379214 58379241 58379421
58392147 58392174 58392417 58392471 58392714 58392741 58394217 58394271 58394721 58397214 58397241 58397421 58732149 58732194
58732419 58732491 58732914 58732941 58734219 58734291 58734921 58739214 58739241 58739421 58793214 58793241 58793421 58932147
58932174 58932417 58932471 58932714 58932741 58934217 58934271 58934721 58937214 58937241 58937421 58973214 58973241 58973421

c)



d)



Aufgabe T6

Gegeben ist ein Array mit dem Inhalt $2, 3, 6, 8, 12, 23, 35, 37, 67, 80, 82, 99$.

Wieviele Vergleiche werden bei einer binären Suche nach den Schlüsseln 1, 2, 8, 23, 24, 81 bzw. 100 durchgeführt?

Lösungsvorschlag

Binsearch(1):

1. $23 > 1$? $(a[m] > x)$
2. $6 > 1$? $(a[m] > x)$
3. $2 > 1$? $(a[m] > x)$

Es wurden 3 Vergleiche durchgeführt

Binsearch(12):

1. $23 > 12$? $(a[m] > x)$
2. $6 < 12$? $(a[m] < x)$
3. $8 < 12$? $(a[m] < x)$
4. $12 == 12$? $(a[m] == x)$

Es wurden 4 Vergleiche durchgeführt

Binsearch(8):

1. $23 > 8$? $(a[m] > x)$
2. $6 < 8$? $(a[m] < x)$
3. $8 == 8$? $(a[m] == x)$

Es wurden 3 Vergleiche durchgeführt

Binsearch(23):

1. $23 == 23$? $(a[m] == x)$

Es wurden 1 Vergleiche durchgeführt

Binsearch(24):

1. $23 < 24$? $(a[m] < x)$

2. $67 > 24$? $(a[m] > x)$
3. $35 > 24$? $(a[m] > x)$

Es wurden 3 Vergleiche durchgeführt

Binsearch(81):

1. $23 < 81$? $(a[m] < x)$
2. $67 < 81$? $(a[m] < x)$
3. $82 > 81$? $(a[m] > x)$
4. $80 < 81$? $(a[m] < x)$

Es wurden 4 Vergleiche durchgeführt

Binsearch(100):

1. $23 < 100$? $(a[m] < x)$
2. $67 < 100$? $(a[m] < x)$
3. $82 < 100$? $(a[m] < x)$
4. $99 < 100$? $(a[m] < x)$

Es wurden 4 Vergleiche durchgeführt

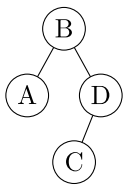
Aufgabe T7
Konstruieren Sie einen optimalen Suchbaum für die Schlüssel A, B, C und D. Auf diese wird mit den Wahrscheinlichkeiten 0.2, 0.3, 0.1 und 0.4 zugegriffen.
Erstellen Sie dazu die Tabellen für $w_{i,j}$ und $e_{i,j}$.

Lösungsvorschlag
Die Tabellen sehen wie folgt aus:

$w_{i,j}$	A	B	C	D
A	0.2	0.5	0.6	1.0
B	0.0	0.3	0.4	0.8
C	0.0	0.0	0.1	0.5
D	0.0	0.0	0.0	0.4

$e_{i,j}$	A	B	C	D
A	0.2(A)	0.7(B)	0.9(B)	1.8(B)
B	0.0	0.3(B)	0.5(B)	1.3(D)
C	0.0	0.0	0.1(C)	0.6(D)
D	0.0	0.0	0.0	0.4(D)

Daraus entsteht folgender optimaler Suchbaum.



Genauere Erklärung: Die erste Tabelle enthält in Zeile i, Spalte j den Wert für $w_{i,j}$, was als $\sum_{k=i}^{j} p_k$ in der Vorlesung definiert wurde. Das heißt, ein Baum, der die Schlüssel i bis j enthält, wird im Erwartungswert bei einer Suche nach einem zufälligen Schlüssel $w_{i,j}$ mal besucht.
In der Vorlesung wurde präsentiert, wie man die Werte dieser ersten Tabelle effizienter mit Hilfe von dynamischer Programmierung berechnen kann. Hier wird einfach nur ausgenutzt, dass wenn man den Wert für $w_{i,j-1}$ hat, $w_{i,j} = w_{i,j-1} + p_j$ ist. Dies folgt direkt aus der Definition von $w_{i,j}$.
Wir wollen jetzt mit Hilfe dieser Tabelle einen optimalen Suchbaum berechnen. Wir definieren $e_{i,j}$ als den Erwartungswert der Anzahl der Vergleiche in einem optimalen Suchbaum, der die Schlüssel von i bis j enthält. Sei n die Anzahl der Schlüssel. Wir wollen den Baum finden, für den $e_{1,n}$ minimal ist, da $e_{1,n}$ die Anzahl der erwarteten Vergleiche in einem Baum, der alle Schlüssel enthält, ist.
Es gilt $e_{i,j} = \min_{i \leq r \leq j}(e_{i,r-1} + e_{r+1,j}) + w_{i,j}$: Die Summe $(e_{i,r-1} + e_{r+1,j}) + w_{i,j}$ drückt für ein bestimmtes r aus, dass die erwartete Anzahl an Vergleichen für einen optimalen Suchbaum, der die Schlüssel von i bis j enthält, *wobei der Schlüssel r der Wurzelknoten ist*, die Summe folgender Teile ist:

- erwartete Anzahl für die Besuche des Wurzelknotens r $(w_{i,j})$
- optimale erwartete Anzahl an Vergleichen im linken Unterbaum $(e_{i,r-1})$
- optimale erwartete Anzahl an Vergleichen im rechten Unterbaum $(e_{r+1,j})$

Es ist wichtig hier anzumerken, dass wenn $j < i$ ist, $e_{i,j} = 0$ ist, da dies ein leerer Unterbaum ist. Wenn wir das Minimum über alle möglichen Wurzelknoten wählen, ist das Ergebnis der optimale Wert $e_{i,j}$.
Wir berechnen $e_{1,n}$ wie folgt: Der optimale Suchbaum mit einem einzigen Knoten ist der Knoten selbst. Es gilt also, dass $e_{i,i} = w_{i,i}$, da ein Baum mit einem Knoten im Erwartungswert genau so oft besucht wird wie der Knoten im Erwartungswert besucht wird. Jetzt können wir mit Hilfe der Formel $e_{i,j} = \min_{i \leq r \leq j}(e_{i,r-1} + e_{r+1,j}) + w_{i,j}$ alle Werte berechnen, wenn $|i - j| = 2$ ist, also für Bäume, die zwei Schlüssel enthalten, da wir alle Werte $e_{i,j}$ kennen, wenn $i = j$ ist. Wir können weiterhin die Werte für Bäume mit immer mehr Knoten berechnen, da wir in der Formel nur die Werte für Bäume brauchen, die weniger Schlüssel enthalten. Jedesmal können wir auch in der Tabelle vermerken, welchen Knoten wir als Wurzelknoten gewählt haben, als wir das Minimum genommen haben. Letztendlich werden wir den optimalen Wert für alle Schlüssel $e_{1,n}$ berechnen.
Wir können jetzt aus der Tabelle den optimalen Suchbaum auslesen: In der Zelle für den Wert $e_{1,n}$ steht die Wurzel r des Baums. Da er ein Suchbaum ist, wissen wir, dass der linke Unterbaum die Schlüssel von 1 bis $r - 1$ enthält, und der rechte Unterbaum die Schlüssel von $r + 1$ bis n. In der Tabelle finden wir dann in der Zelle für $e_{1,r-1}$ die Wurzel vom linken Unterbaum und in $e_{r+1,n}$ die Wurzel vom rechten Unterbaum. Wir können nach dieser Logik dann den Baum rekursiv aufbauen, indem wir immer in der Tabelle nachgucken, was die nächsten Wurzeln sind. Das Ergebnis ist dann der optimale Suchbaum.

Aufgabe T8

Am Bartresen einer Kneipe in der Pontstraße spricht Sie ein Unbekannter an. Er behauptet: Wenn man in einen zu Beginn leeren binären Suchbaum die gleichen Elemente in unterschiedlichen Reihenfolgen einfügt, dann sind die entstehenden binären Suchbäume danach verschieden. Finden Sie einen Beweis oder ein Gegenbeispiel für diese Behauptung. Nach Möglichkeit sollten sie nicht mehr Platz verwenden, als ein handelsüblicher Bierdeckel bietet.

Lösungsvorschlag

Wenn man die Elemente 1, 5, 8 in den Reihenfolgen 5, 1, 8 bzw. 5, 8, 1 in einen leeren binären Suchbaum einfügt, entsteht jeweils der gleiche Baum. Damit ist die Behauptung widerlegt.

Aufgabe H4 (10 Punkte)
In der Vorlesung wurde binäre Suche analysiert. Wir müssen nun noch die letzte Lücke schließen und

$$\lfloor \log n \rfloor = \left\lfloor \log \lceil (n-1)/2 \rceil \right\rfloor + 1$$

für $n > 1$ beweisen. Mit $\lfloor$ und $\lceil$-en rechnen zu müssen ist eine typische Schwierigkeit beim Entwurf von Algorithmen. Beweisen Sie, dass obige Aussage gilt.

Lösungsvorschlag
Sei $k \geq 0$, so dass entweder $n = 2k$ oder $n = 2k+1$. In beiden Fällen ist $\lceil (n-1)/2 \rceil = k$. Dann ist

$$\left\lfloor \log \lceil (n-1)/2 \rceil \right\rfloor + 1 = \left\lfloor \log k + \log 2 \right\rfloor = \left\lfloor \log 2k \right\rfloor$$

Für $n = 2k$ ist die Aussage damit bereits gezeigt. Für ungerade $n = 2k+1$ sei $q \geq 0$ maximal mit $2^q < n$, also $q = \lfloor \log n \rfloor$. Wegen $2^q \leq 2k$ gilt dann auch

$$q \leq \lfloor \log 2k \rfloor \leq \lfloor \log n \rfloor = q.$$

Aufgabe H5 (10 Punkte)
Konstruieren Sie einen optimalen Suchbaum für die Wörter (und zugehörigen Zugriffswahrscheinlichkeiten) RWTH(0.10), ETH(0.25), KIT(0.10), Stanford(SU)(0.35), und TUM(0.20) bezüglich der lexikographischen Ordnung.
Erstellen Sie die Tabellen für $w_{i,j}$ und $e_{i,j}$. Geben Sie den resultierenden, optimalen Suchbaum graphisch an. Ist dieser eindeutig?

Lösungsvorschlag
Die Tabellen sehen wie folgt aus:

$w_{i,j}$	ETH	KIT	RWTH	SU	TUM
ETH	0.25	0.35	0.45	0.80	1.00
KIT	0.00	0.10	0.20	0.55	0.75
RWTH	0.00	0.00	0.10	0.45	0.65
SU	0.00	0.00	0.00	0.35	0.55
TUM	0.00	0.00	0.00	0.00	0.20

$e_{i,j}$	ETH	KIT	RWTH	SU	TUM
ETH	0.25 (ETH)	0.45 (ETH)	0.75 (ETH)	1.55 (SU)	1.95 (SU)
KIT	0.00	0.10 (KIT)	0.30 (KIT,RWTH)	0.85 (SU)	1.25 (SU)
RWTH	0.00	0.00	0.10 (RWTH)	0.55 (SU)	0.95 (SU)
SU	0.00	0.00	0.00	0.35 (SU)	0.75 (SU)
TUM	0.00	0.00	0.00	0.00	0.20 (TUM)

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

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

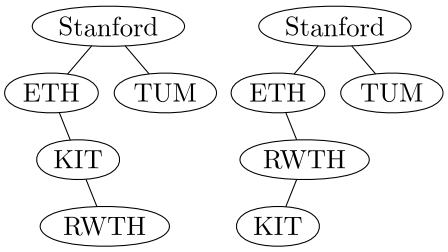
In den Klammern hinter den Werten steht, welches r jeweils gewählt wurde, um diesen Wert zu erreichen. Beispielsweise gilt für $e_{\text{ETH,RWTH}}$:

$$\begin{aligned} e_{\text{ETH,RWTH}} &= \min_{\text{ETH} \leq r \leq \text{RWTH}} (e_{\text{ETH},r-1} + e_{r+1,\text{RWTH}}) + w_{\text{ETH,RWTH}} \\ &= \min(e_{\text{ETH,ETH-1}} + e_{\text{KIT,RWTH}}, e_{\text{ETH,ETH}} + e_{\text{RWTH,RWTH}}, e_{\text{ETH,KIT}} + e_{\text{SU,RWTH}}) + 0.45 \\ &= \min(0 + 0.30, 0.25 + 0.10, 0.45 + 0) + 0.45 \\ &= 0.3 + 0.45 \end{aligned}$$

Hier wurde $r = \text{ETH}$ gewählt, um den minimalen Wert 0.75 zu erhalten, also muss die Wurzel des Teilbaums mit allen Schlüsseln zwischen ETH und RWTH (also den Schlüsseln ETH, KIT und RWTH) der Knoten mit dem Schlüssel ETH 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.

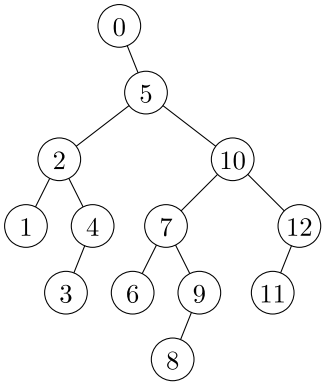
Wir lesen nun die Struktur des optimalen Suchbaums ab. Dabei werden wir im folgenden Gleichheitszeichen verwenden, wenn wir den Index einer Zeile oder Spalte auf den ihr zugeordneten Namen abbilden.

Wir können in $e_{1,5} = e_{\text{ETH,TUM}}$ ablesen, dass die Wurzel des optimalen Suchbaums Stanford (SU) sein muss. Wenn wir also $r = \text{SU}$ setzen, können wir die Wurzel des linken Teilbaums in $e_{1,r-1} = e_{\text{ETH,RWTH}}$ ablesen, diese lautet ETH. Die Wurzel des rechten Teilbaums finden wir entsprechend in $e_{r+1,5} = e_{\text{TUM,TUM}}$, es handelt sich dabei um TUM. Die Wurzel des linken Teilbaums von ETH können wir wiederrum in dem Feld $e_{1,r-1} = e_{1,0}$ finden, dieser Teilbaum ist also leer. Die Wurzel des rechten Teilbaums von ETH finden wir in dem Feld $e_{r+1,3} = e_{\text{KIT,RWTH}}$, hier finden wir die Elemente KIT und RWTH. An dieser Stelle ist der optimale Suchbaum demnach nicht eindeutig, wir erhalten den optimalen Wert unabhängig davon, welchen der beiden Knoten wir als Wurzel wählen. Der andere Knoten ist auf Grund der lexikographischen Reihenfolge entsprechend das Kind des anderen Knoten. Somit ergeben sich zwei optimale Suchbäume:



Aufgabe H6 (10 Punkte)

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.

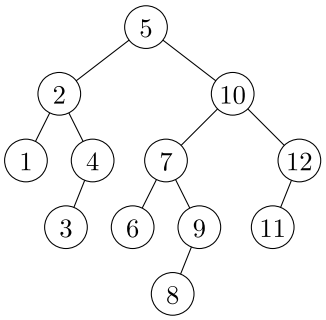
b) Löschen Sie die 10.

c) Fügen Sie die 10 ein.

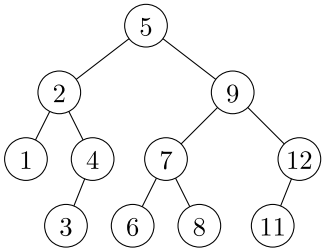
d) Löschen Sie die 5.

Lösungsvorschlag

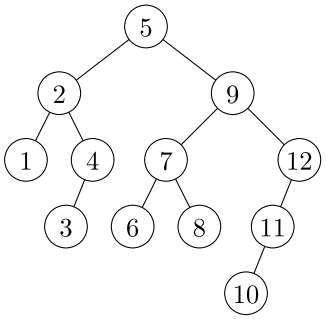
Löschen Sie die 0:



Löschen Sie die 10:



Fügen Sie die 10 ein:



Löschen Sie die 5: