Optimale Suchbäume

Anzahl Zugriffe:

	A		**B**		**C**		**D**		**E**	
	1		3		3		3		4	
(-∞,A)		**(A,B)**		**(B,C)**		**(C,D)**		**(D,E)**		**(E,∞)**
4		1		0		2		0		10

Gewichtsmatrix W:

Annahme für Feld W[i,j]:

Teilbaum von (i,..) bis (..,j) hängt ganz unten im optimalen Suchbaum.

Wieviele Zugriffe auf diesen ganzen Teilbaum finden statt?
-> Aufsummieren aller Zugriffe dazwischen

Z.B. für Teilbaum in Feld W[B,D]:

W[B,D]:= (B,C) + C + (C,D) + D + (D,E) = 0+3+2+3+0 = 8

Vorsicht: Knoten B gehört hier nicht dazu! -> **die Zeilenbezeichnung bezieht sich immer auf den Zwischenraum links neben dem entsprechenden Knoten**

	-	**A**	**B**	**C**	**D**	**E**
-						
A						
B						
C						
D						
E						

Hier muss die Summe über alle Zugriffe stehen, da der Teilbaum W[-,E] ja alle Knoten und Zwischenräume enthält

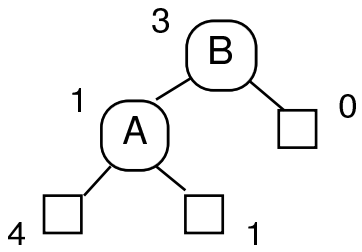
Pfadlängenmatrix P:

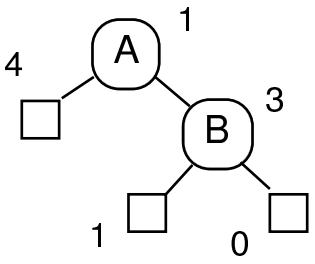
Wie kann die Wurzel für einen bestimmten Teilbaum möglichst optimal gewählt werden. Durch die Wahl der Wurzel werden für die 2 Teile links und rechts der Wurzel die Zugriffspfade je um einen Schritt länger -> Suche die Wurzel, bei der sich die Länge der Zugriffspfade um möglichst wenig vergrössert.

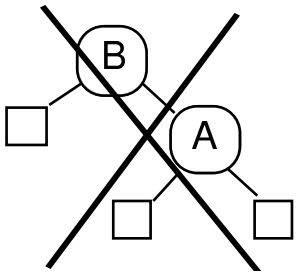
Z.B. für Feld P[-,B]:

Mögliche Wurzeln sind die Knoten A und B. Falls A als Wurzel gewählt wird, so werden die Zugriffspfade für Teilbaum [A,B] um 1 länger, für die Wahl von B die Zugriffspfade von [-,A].

Mögliche entstehende Bäume:







Nicht möglich, muss Suchbaum sein!

Für Wurzel A würden für 4 Zugriffe die Pfadlänge um 1 vergrössert, für die Wurzel B 6 Zugriffe, das heisst A ist die bessere Wurzel.

	-	A	B	C	D	E
-	0	6	13			
A		0	4			
B			0			
C				0		
D					0	
E						0

Diagonale mit Wert 0 initialisieren. Dann für Feld P[i,j]: Für alle möglichen Wurzeln l testen:

P[i,j]:= min (P[i,l-1],P[l,j]) + W[i,j]

Beim Ausfüllen dieser Pfadmatrix wird parallel dazu noch die dritte Matrix R ausgefüllt.
In dieser Matrix wird für jedes Feld eingetragen, welches die optimale Wahl für die Wurzel war.

zB für P[-,B]: Entweder für Wurzel A: 0 + 4 + 9 = 13
oder für Wurzel B: 6 + 0 + 9 = 15

-> Wurzel A wählen, in Matrix R eintragen

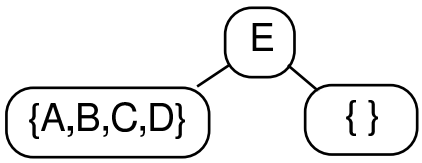
Wurzelmatrix R:

	-	A	B	C	D	E
-		A	A			
A			B			
B				C		
C					D	
D						E
E						

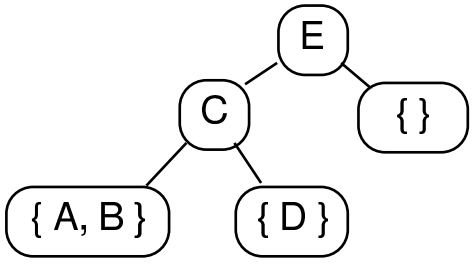
Am Ende der Berechnungen stehen also im Feld rechts oben (für den ganzen Baum [- bis E]) der Pfadmatrix und der Wurzelmatrix die Lösung. Wie kann nun daraus der Baum konstruiert werden?

Aufgrund der Wurzelmatrix wird im Feld rechts oben die Wurzel des gesamten Baumes festgelegt. Dadurch wird die Menge der Knoten in 3 Teile geteilt: Den Teilbaum links unten der Wurzel, die Wurzel selber und den Teilbaum rechts unten. Die Teilbäume können auch die leere Menge sein.

zB steht bei uns im Feld [-,E] der Knoten E -> E ist die Wurzel des ganzen Baumes.



Nun wird für die entsprechenden Teilbäume wieder die Wurzel gesucht. Für den linken Teilbaum steht sie in Feld [-,D] (das heisst, C ist die Wurzel dieses Teilbaums), für den linken Teilbaum gibt es keine neue Wurzel.

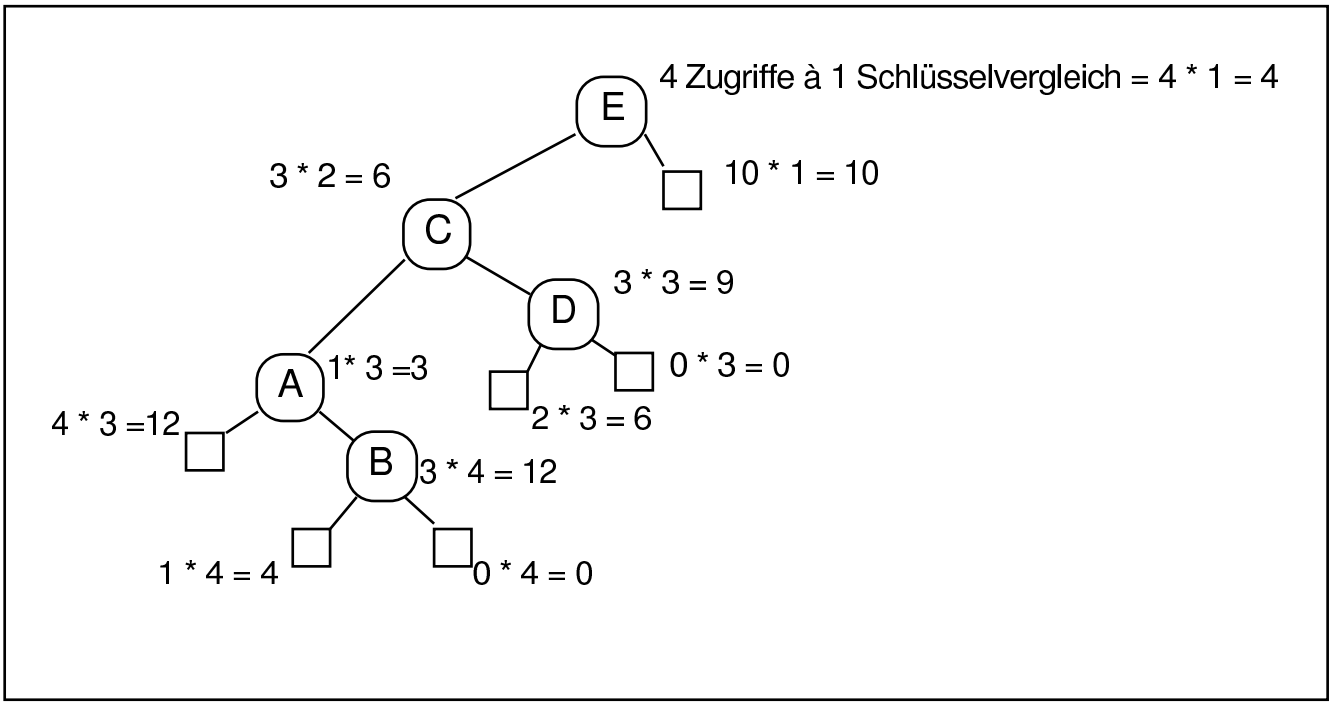


So wird der ganze Baum aufgebaut.

Nun kann noch die Summe der Pfadlängen für alle Zugriffe berechnet werden.

Dabei ist zu beachten, dass nur Schlüsselvergleiche gezählt werden. Das heisst, wenn ich in Knoten A bin und auf einen Schlüssel kleiner als A zugreifen will und dort nur NIL finde, wie das im Ganzen ja 4 Mal vorkommt, so wird für die Pfadlänge zu (-∞,A) nur die gleiche Pfadlänge gezählt wie zu A, da ja kein zusätzlicher Schlüsselvergleich zusätzlich stattfinden muss.

So sieht der konstruierte Baum mit der jeweiligen Anzahl Zugriffe und der benötigten Anzahl Schlüsselvergleiche für jeden Knoten aus:



Die Summe beträgt also 4 + 6 + 10 + 9 + 3 + 6 + 0 + 12 + 12 + 4 + 0 = 66

Das ist genau die Zahl, die im rechten oberen Feld der Pfadängenmatrix P steht. Dort wurde ja die minimale Pfadlänge für den gesamten Baum berechnet (d.h. für den Teilbaum von - bis E berechnet).

Der Baum, der so entstanden ist, ist der optimale Suchbaum. Er minimert also die Anzahl der Schlüsselvergleiche aufgrund der vorgegebenen Häufigkeiten der Zugriffe auf und zwischen die Elemente des Baumes.

Ausgefüllte Matrizen:

Gewichtsmatrix W:

	-	A	B	C	D	E
-	4	6	9	14	17	31
A		1	4	9	12	26
B			0	5	8	22
C				2	5	19
D					0	14
E						10

Pfadlängenmatrix P:

	-	A	B	C	D	E
-	0	6	13	25	35	66
A		0	4	13	21	47
B			0	5	13	35
C				0	5	24
D					0	14
E						0

Wurzelmatrix R:

	-	A	B	C	D	E
-		A	A	B	C	E
A			B	C	C	E
B				C	C / B	E
C					D	E
D						E
E						