

A1

Preorder: Zuerst wird die Wurzel ausgelesen, danach erst der linke und dann der rechte Teilbaum.

➔ Ausgabe: Das ist das beste Modul ,das ich bisher habe belegt !

Inorder: Zuerst wird der linke Teilbaum, dann die Wurzel und am Ende der rechte Teilbaum ausgelesen

➔ Ausgabe: das ist Modul beste Das ,das habe bisher belegt ich !

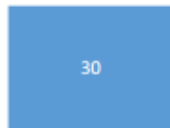
Postorder: Zuerst wird der linke, dann der rechte Teilbaum und am Ende die Wurzel ausgegeben.

➔ Ausgabe: das Modul beste ist habe belegt bisher ! ich ,das Das

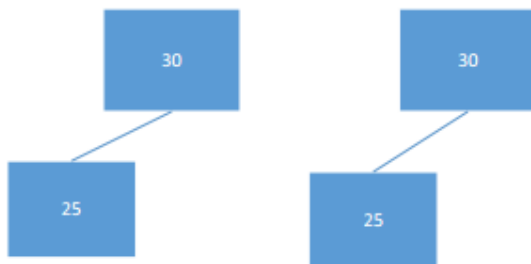
A2. AVL-Baum

- a) Anmerkung: Die Balancierung wurde immer nur jeweils für den Vaterknoten angepasst, der direkt über dem eingefügten Knoten ist. Erst bei dem endgültigem Baum ist bei jedem Schritt für alle Knoten der BF angepasst.

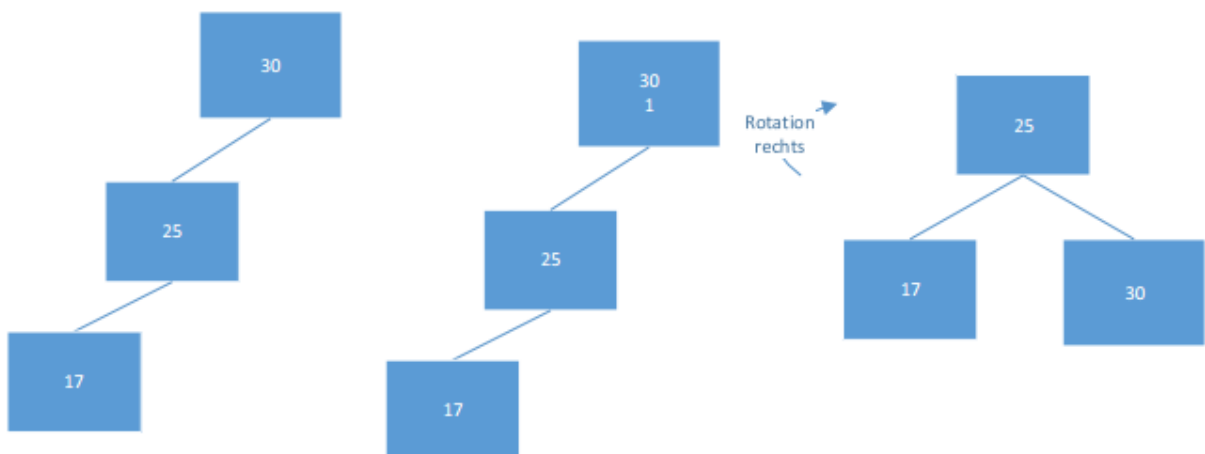
Einfügen 30:



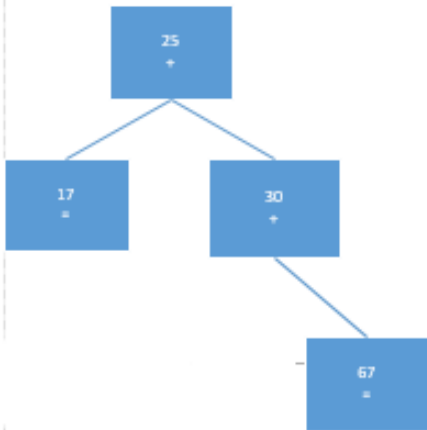
Einfügen 25:



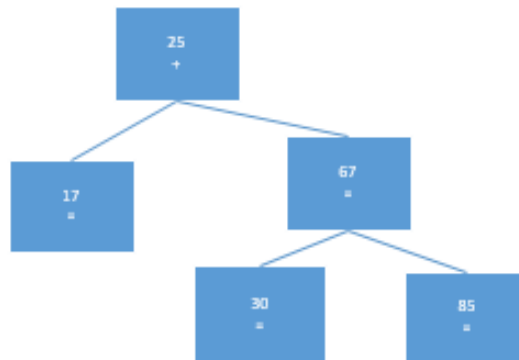
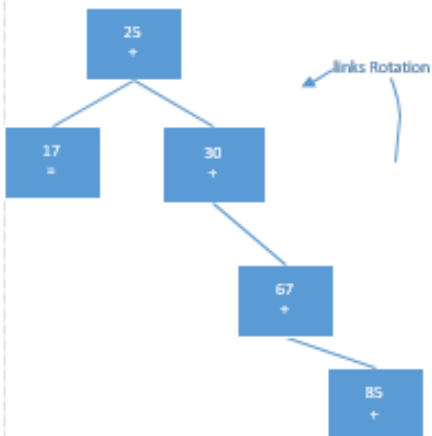
Einfügen 17 :



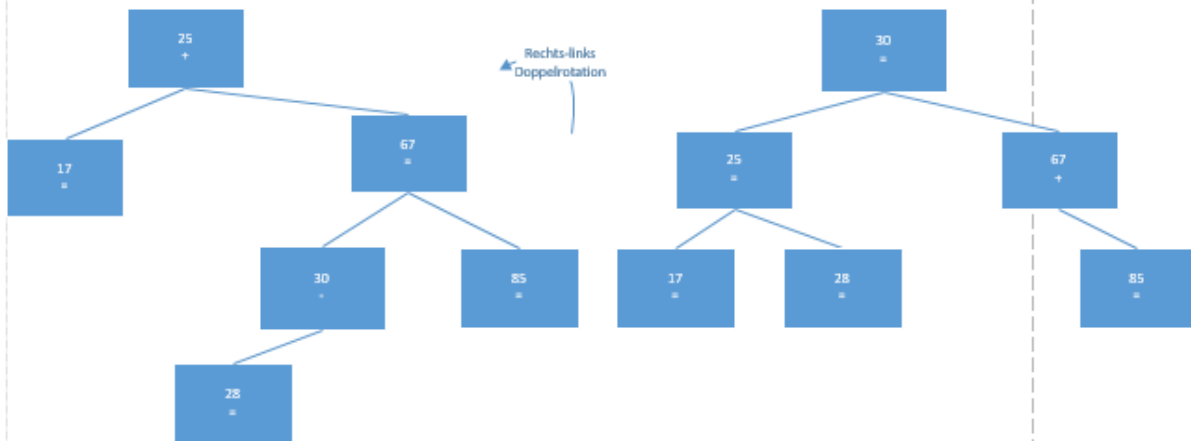
Einfügen 67 :



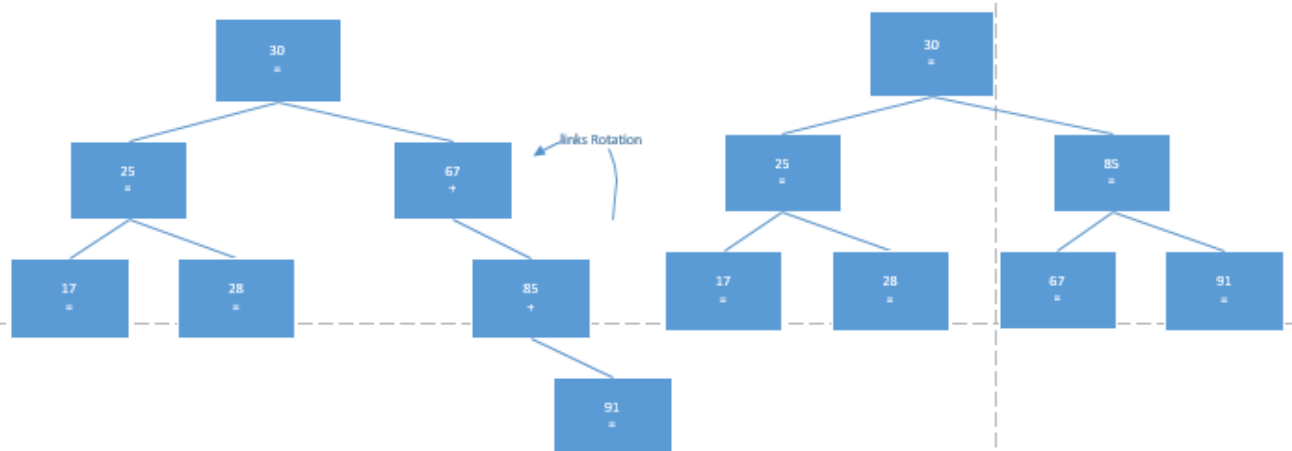
Einfügen 85 :



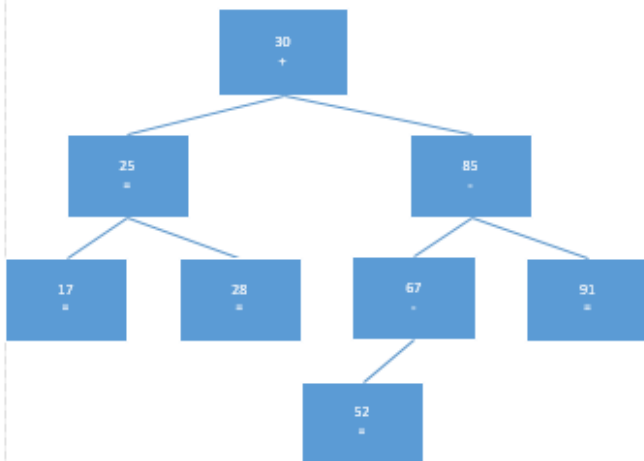
Einfügen 28 :



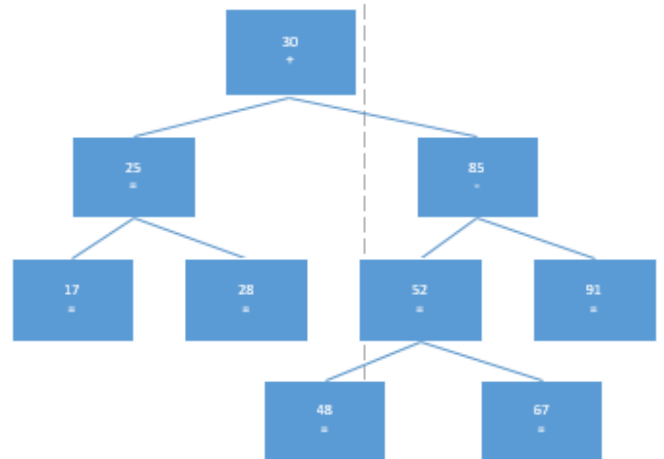
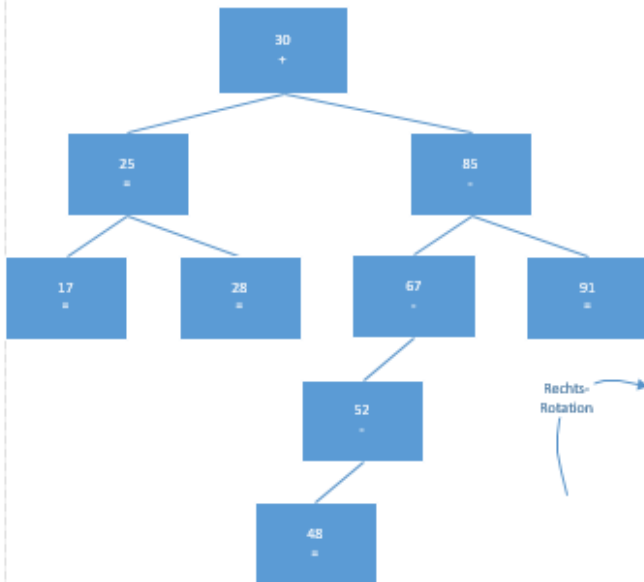
Einfügen 91 :

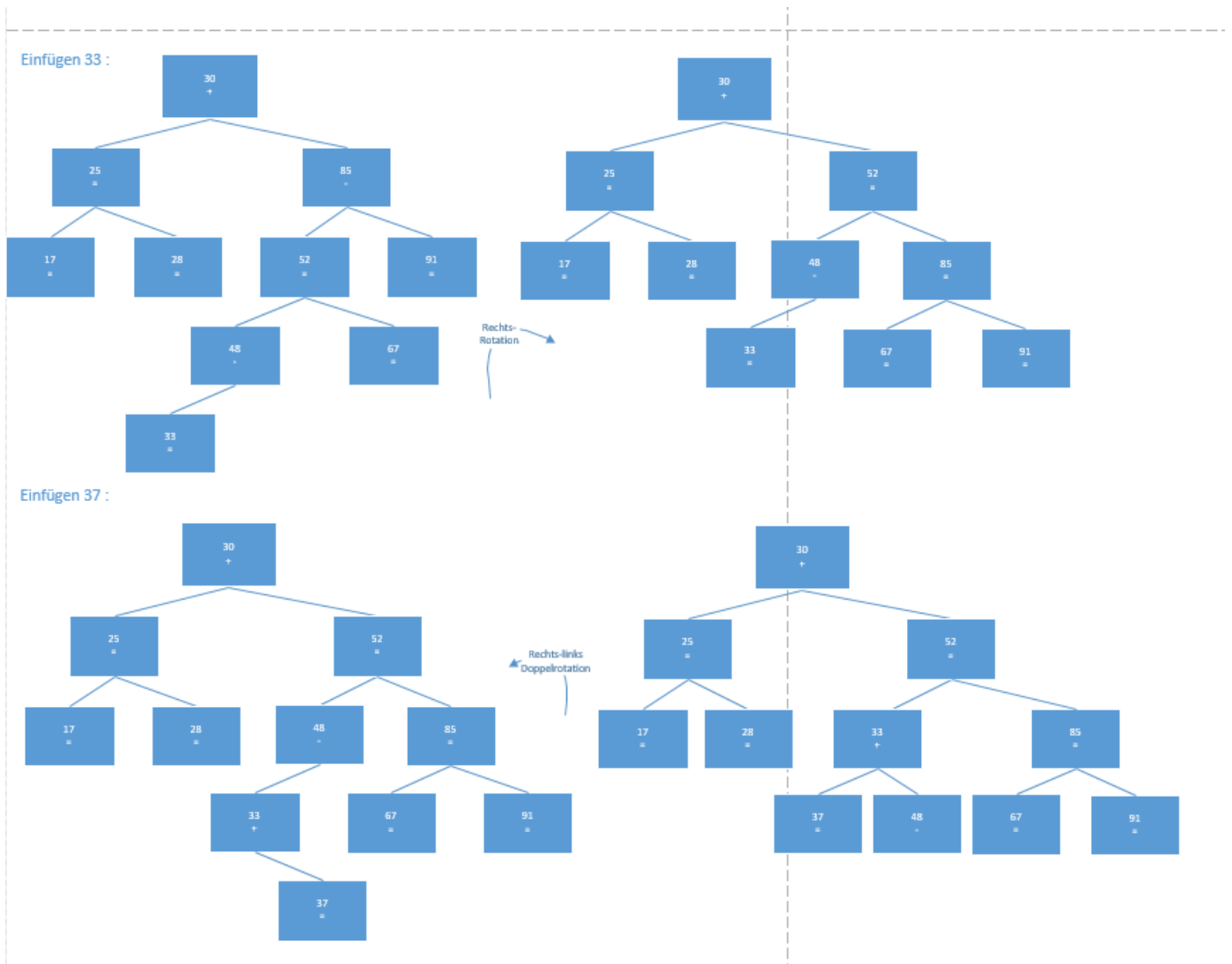


Einfügen 52 :

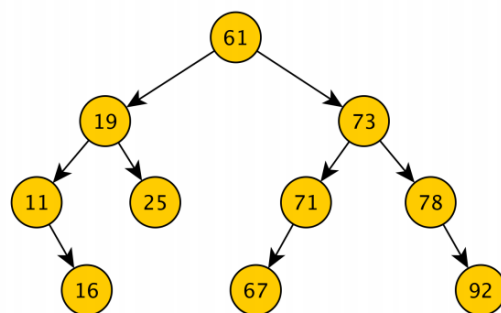


Einfügen 48 :

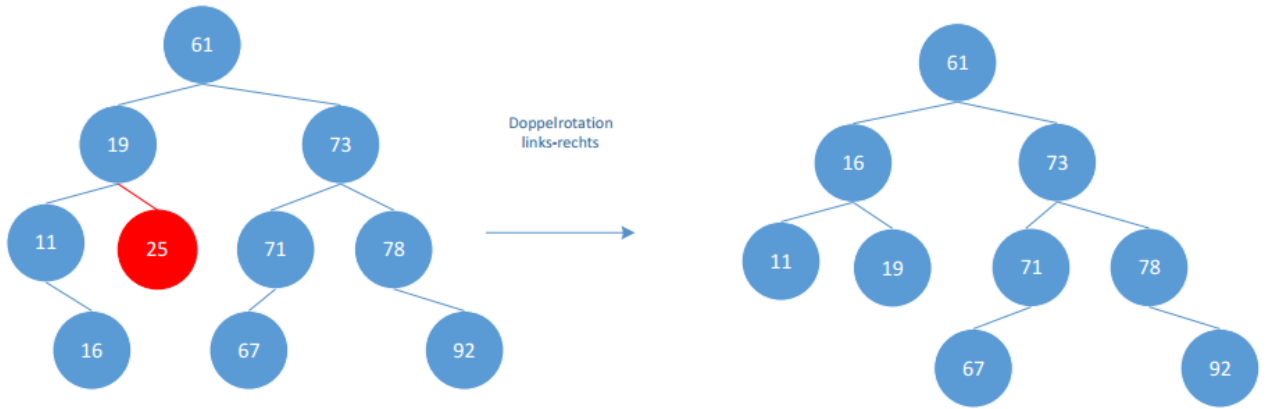




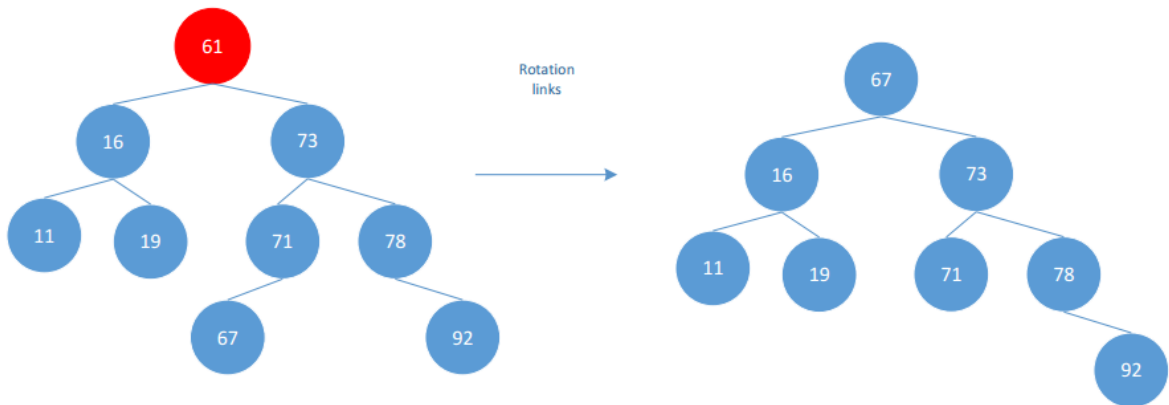
b) Löschen aus AVL-Baum, die Werte 25, 61 u. 16



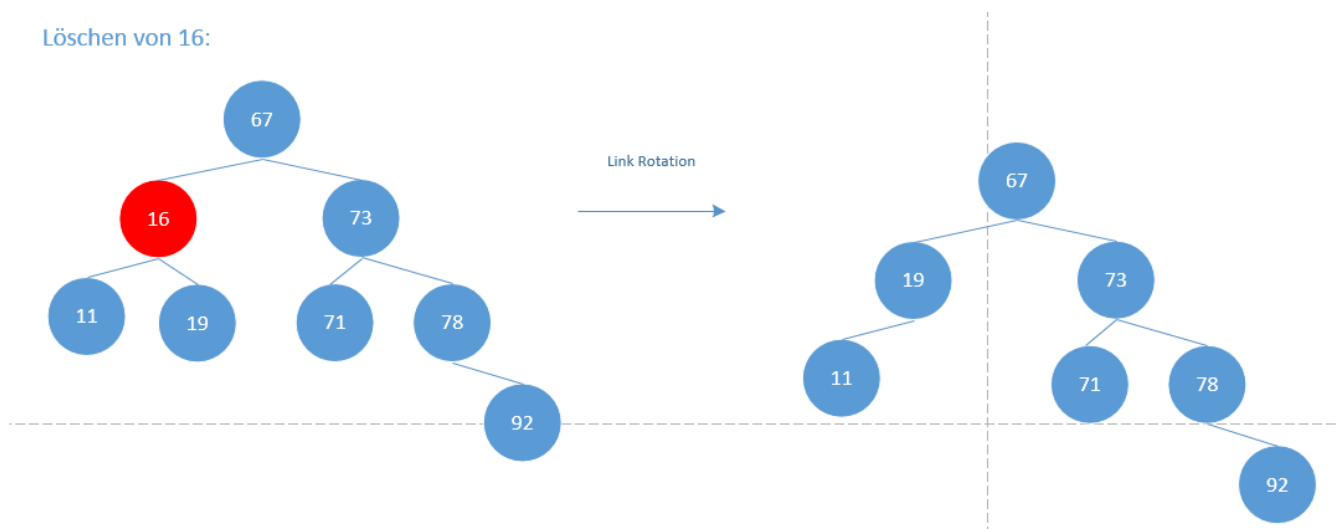
Löschen von 25:



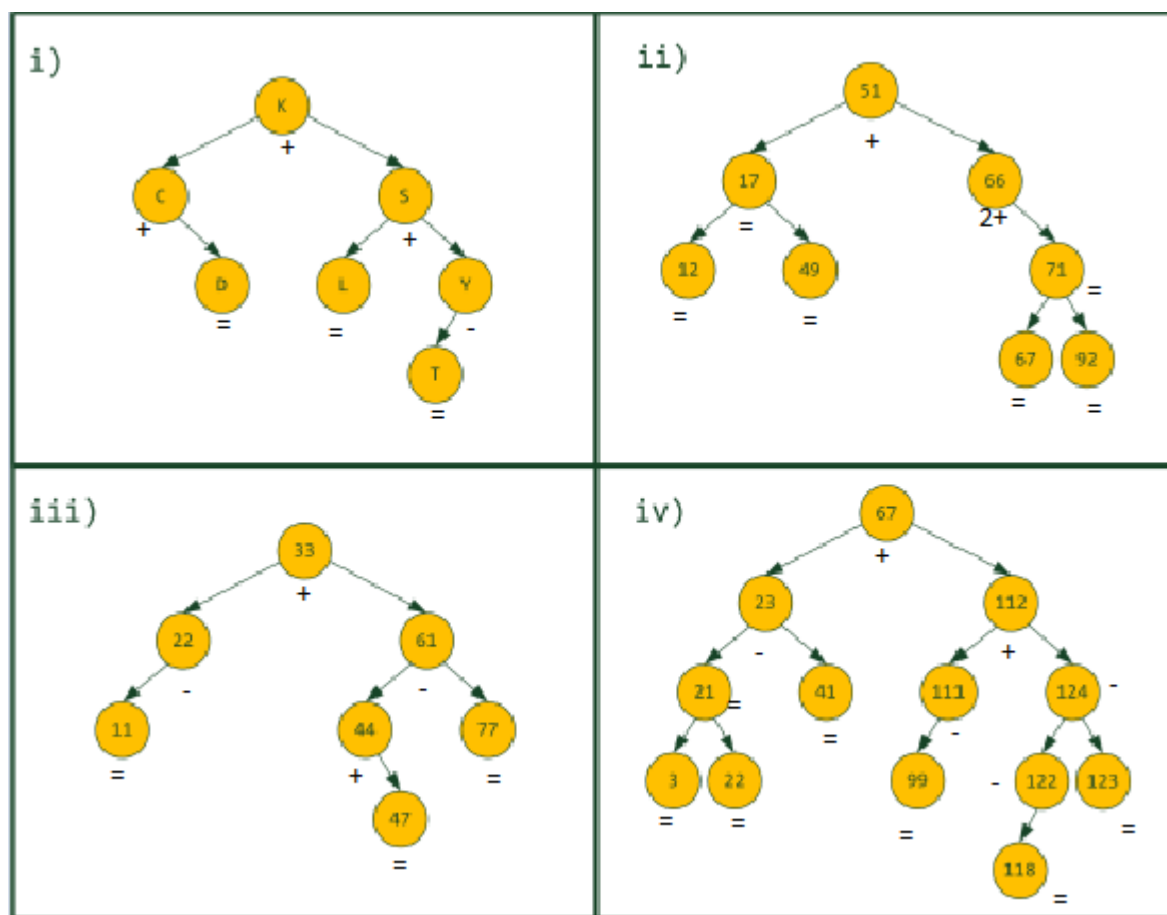
Löschen von 61:



Löschen von 16:



c) Welche Bäume sind AVL-Bäume?

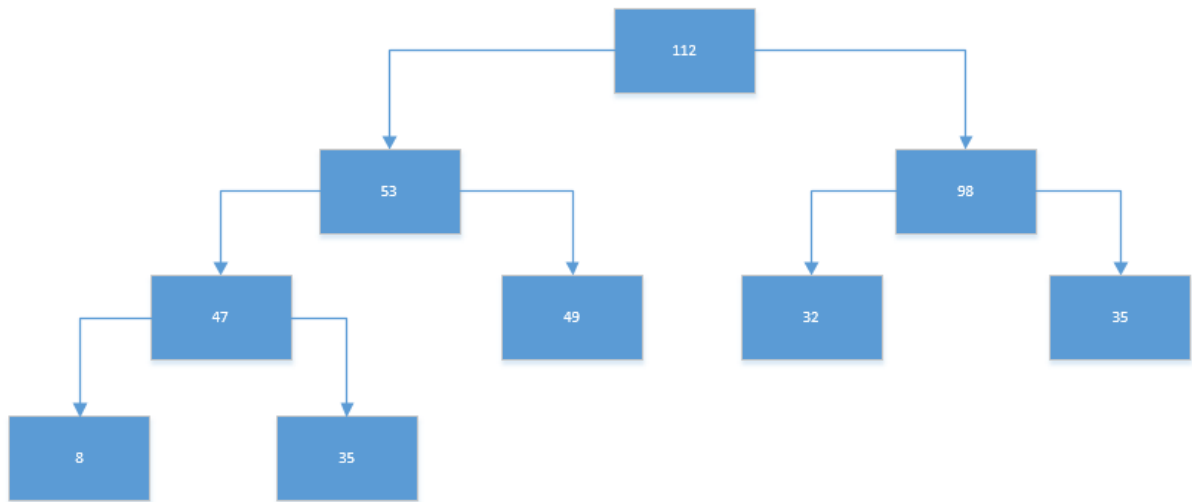


Alle außer ii) sind AVL-Bäume, da sonst der BF immer passt. (Siehe Bild)

A3:

Ausgangssituation:

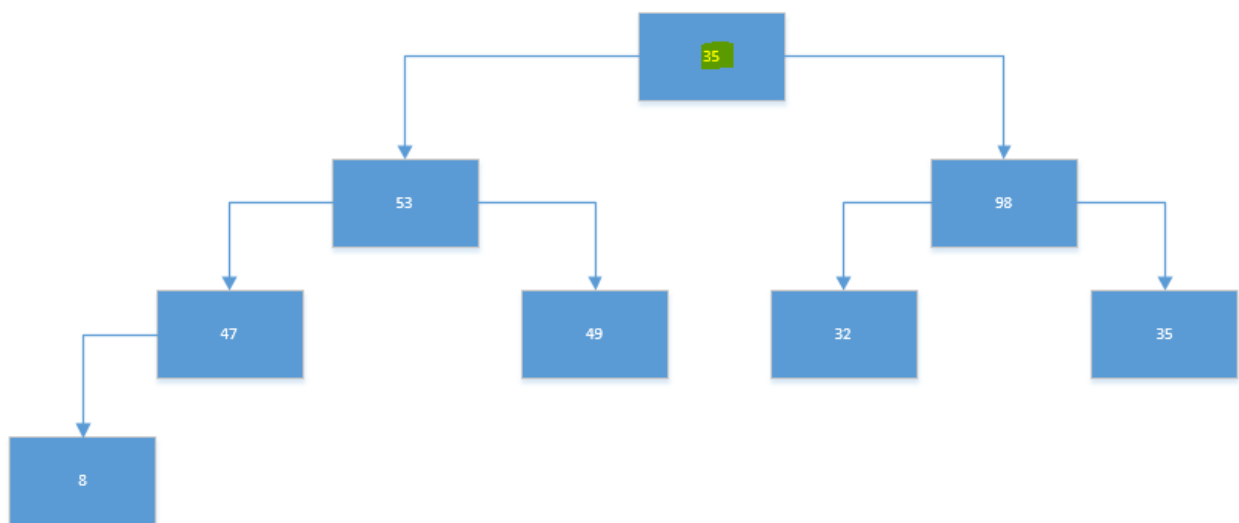
0	1	2	3	4	5	6	7	8	9	10	11
	112	53	98	47	49	32	35	8	35		



Entfernung 112:

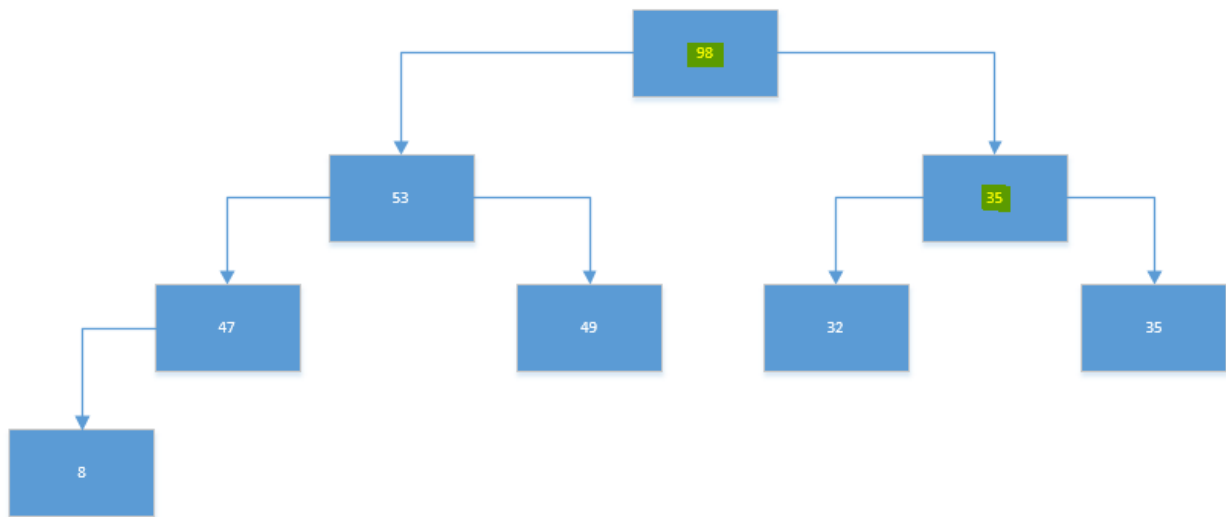
- Einfügen des letzten Elements an die erste Stelle

0	1	2	3	4	5	6	7	8	9	10	11
	35	53	98	47	49	32	35	8			



- Aufruf downheap(1) um 35 möglicherweise abzusenken
- Es existiert ein linker Teilbaum (53), dieser Teilbaum ist größer
 - ➔ $\text{index} = 2i = 2$
 - ➔ $53 < 98 \rightarrow \text{index}++ = 3$
 - ➔ Vertauschen von index (98) und i (35)

0	1	2	3	4	5	6	7	8	9	10	11
	98	53	35	47	49	32	35	8			

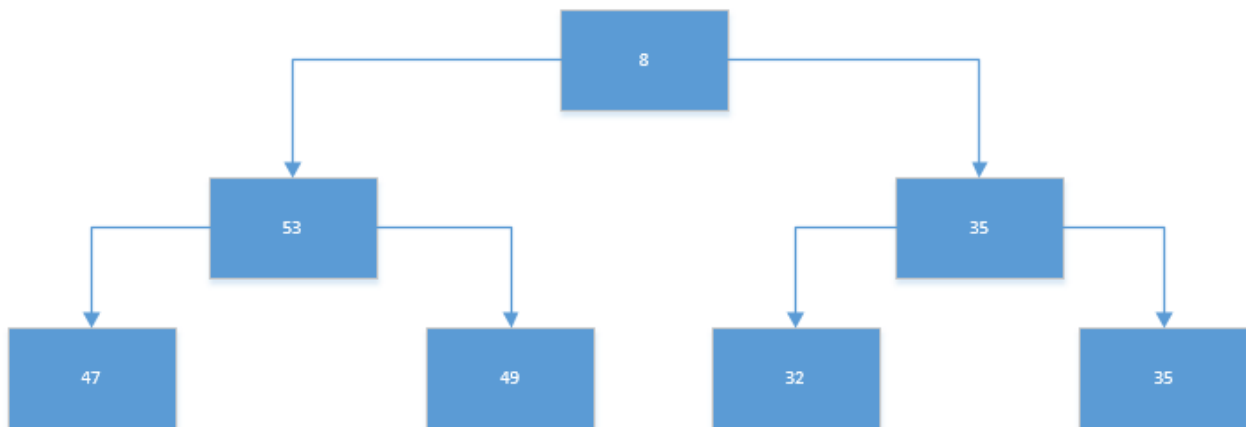


- Aufruf downheap(3)
- Es existiert ein linker Teilbaum (32), dieser Teilbaum ist kleiner und der rechte Teilbaum (35) ist gleich. Jetzt passt alles und die Löschung ist beendet.

Entfernung 98:

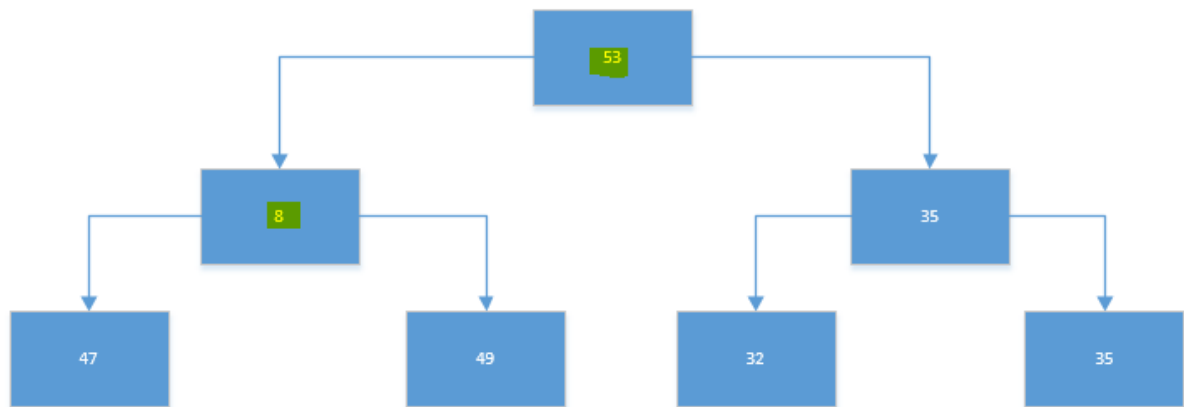
- Einfügen des letzten Elements an der ersten Stelle

0	1	2	3	4	5	6	7	8	9	10	11
	8	53	35	47	49	32	35				



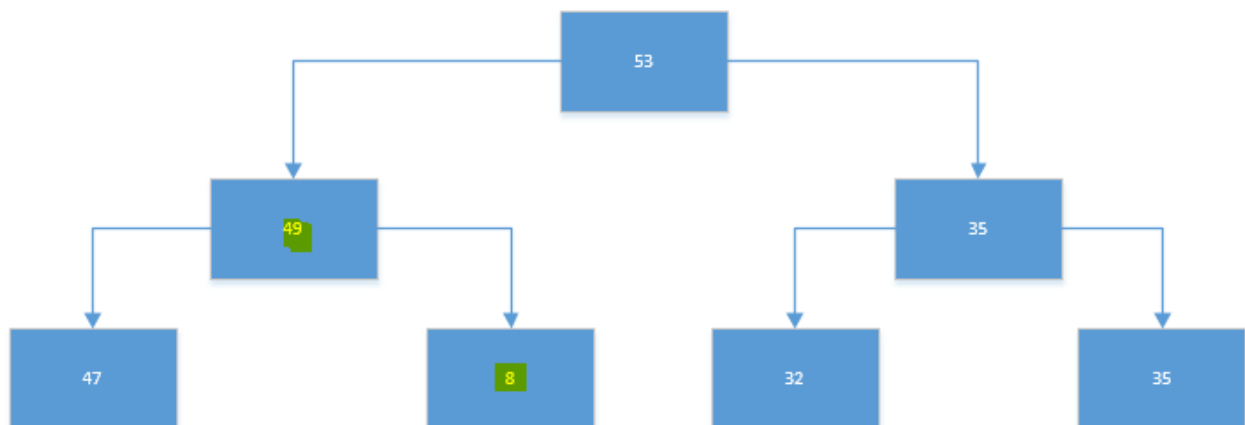
- Aufruf downheap(1) um 8 möglicherweise abzusenken
- Es existiert ein linker Teilbaum (53), der größer ist
- ➔ Index = $2i = 2$
- ➔ $53 > 35 \rightarrow$ nichts passiert
- ➔ Vertauschen von index (8) und i (53)

0	1	2	3	4	5	6	7	8	9	10	11
	53	8	35	47	49	32	35				



- Aufruf downheap(2)
- Es existiert ein linker Teilbaum (47), der größer ist
- ➔ Index = $2i = 4$
- ➔ $47 < 49 \rightarrow \text{index}++ = 5$
- ➔ Vertauschen von index (49) und i (8)

0	1	2	3	4	5	6	7	8	9	10	11
	53	49	35	47	8	32	35				

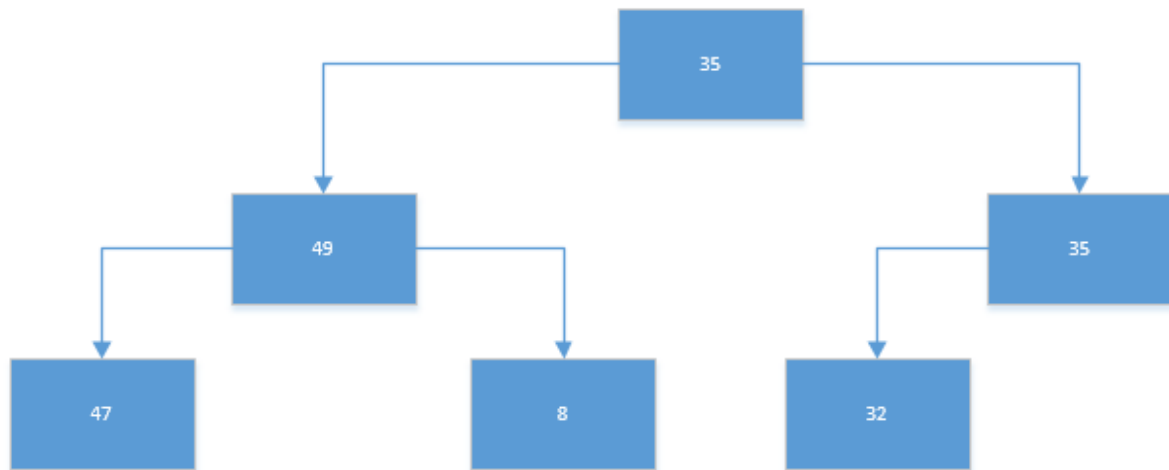


- Aufruf downheap(5)
- Es existiert kein linker Teilbaum -> Ende

Entfernung 53:

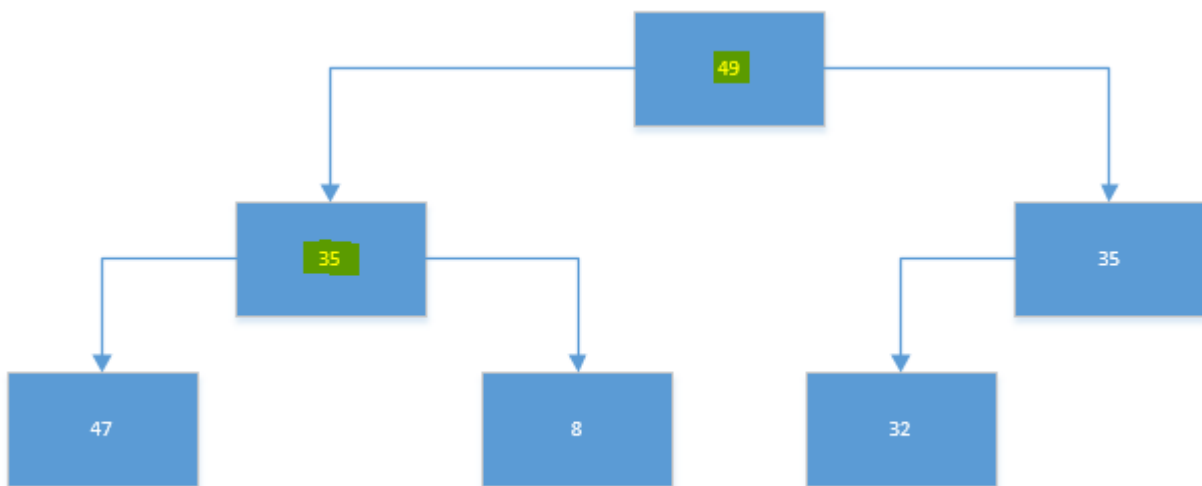
- Einfügen des letzten Elements an der ersten Stelle

0	1	2	3	4	5	6	7	8	9	10	11
	35	49	35	47	8	32					



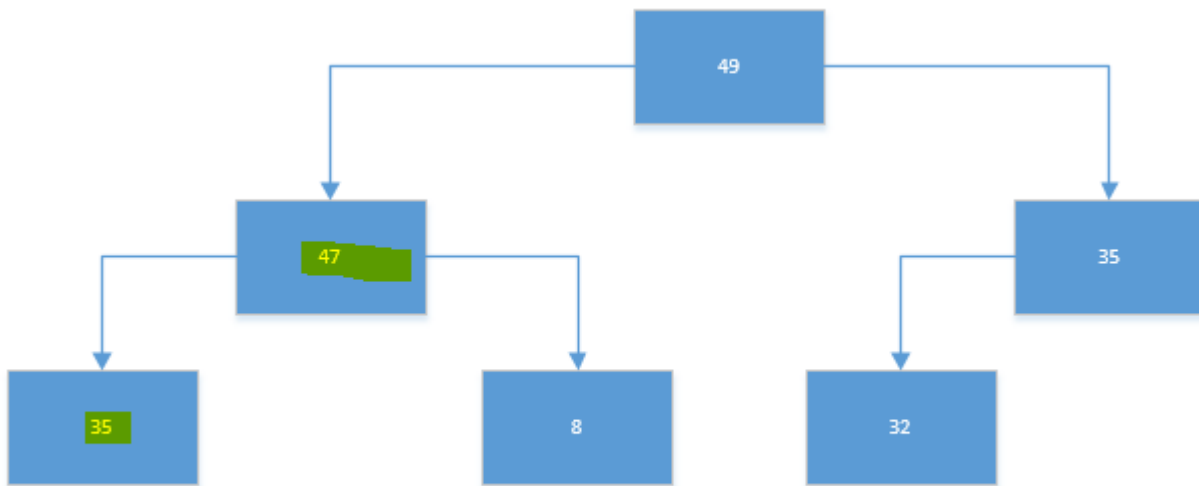
- Es existiert ein linker Teilbaum (49) der größer ist
- ➔ Index = $2i = 2$
- ➔ $49 > 35 \rightarrow$ keine Aktion
- ➔ Vertauschen von index (49) und i (35)

0	1	2	3	4	5	6	7	8	9	10	11
	49	35	35	47	8	32					



- Aufruf downheap(2)
- Es existiert ein linker Teilbaum, der größer ist
- ➔ Index = $2i = 4$
- ➔ $47 > 8 \rightarrow$ keine Aktion
- ➔ Vertauschen von index (47) und i (35)

0	1	2	3	4	5	6	7	8	9	10	11
	49	35	47	35	8	32					

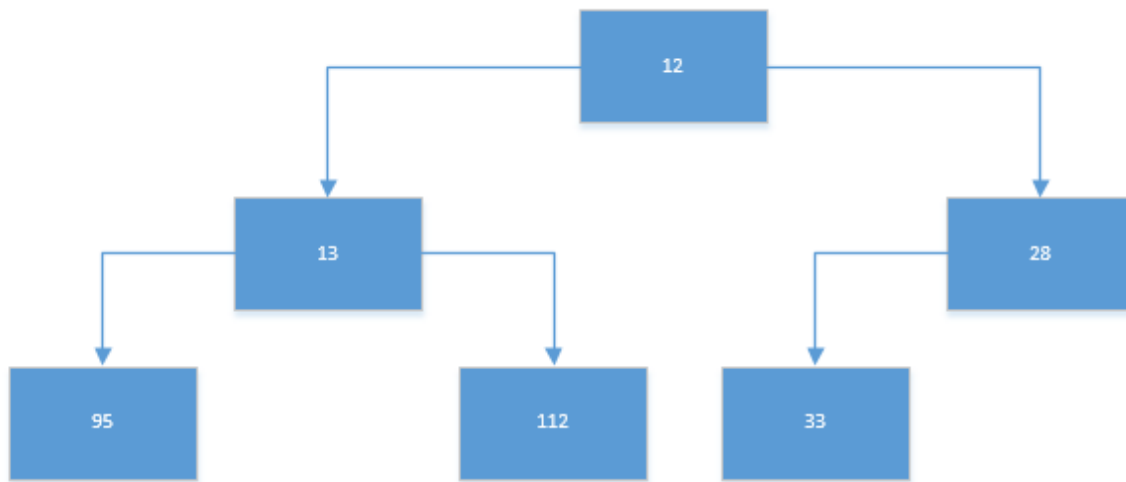


- Aufruf downheap(4)
- Es existieren keine Teilbäume -> Ende

b)

Ausgangssituation:

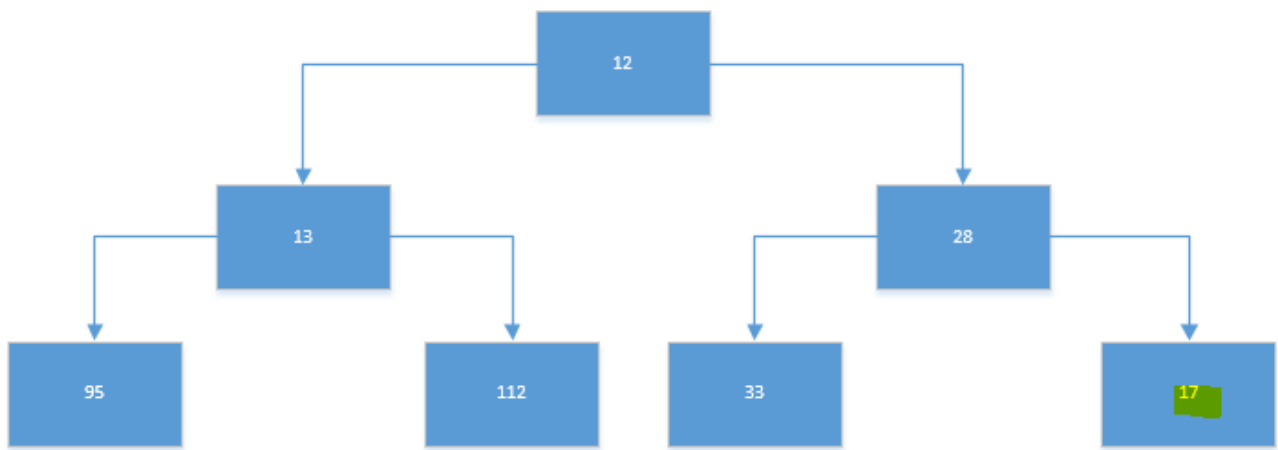
0	1	2	3	4	5	6	7	8	9	10	11	12	13
	12	13	28	95	112	33							



Einfügen 17:

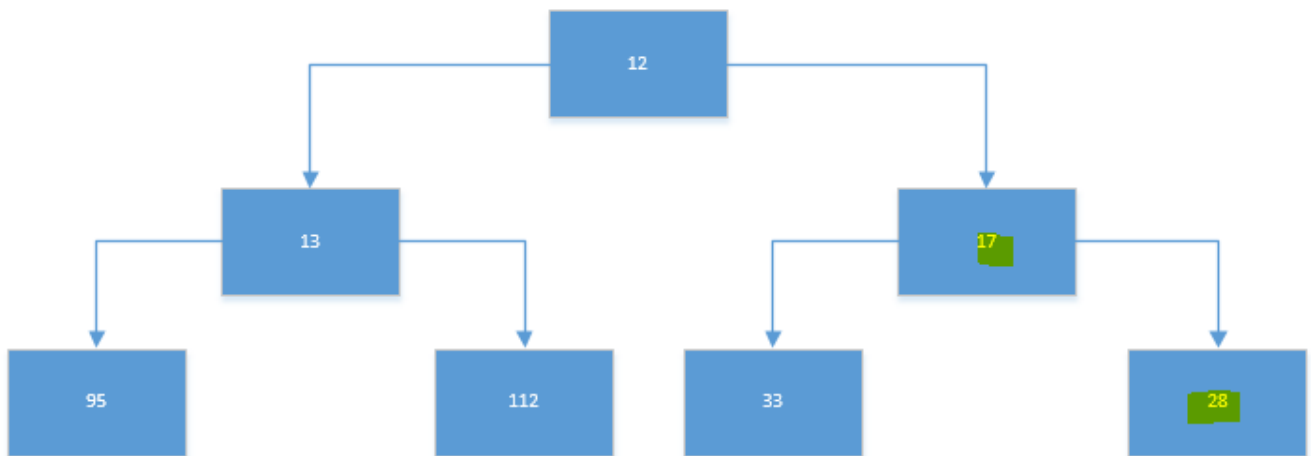
- Einfügen am Ende

0	1	2	3	4	5	6	7	8	9	10	11	12	13
	12	13	28	95	112	33	17						



➤ $17 < 28 \rightarrow$ Vertauschen

0	1	2	3	4	5	6	7	8	9	10	11	12	13
	12	13	17	95	112	33	28						

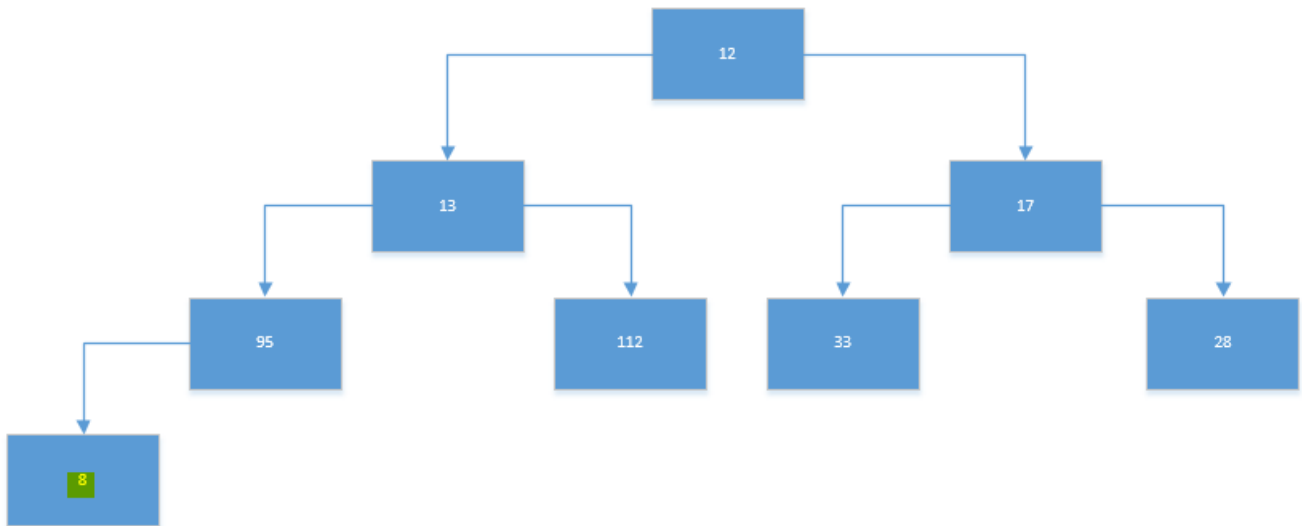


➤ $17 > 12 \rightarrow$ Ende

Einfügen von 8:

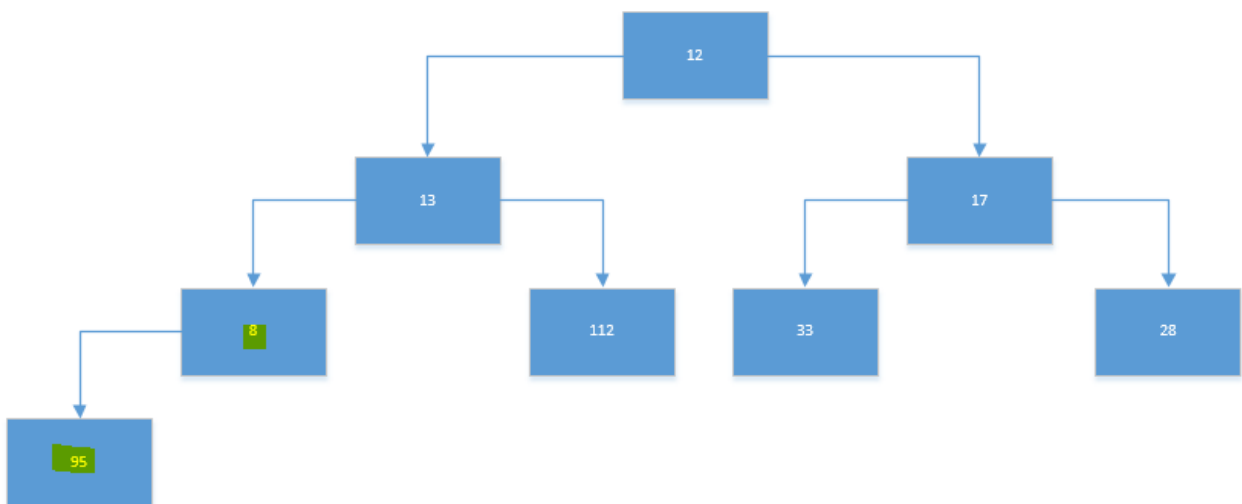
➤ Einfügen am Ende

0	1	2	3	4	5	6	7	8	9	10	11	12	13
	12	13	17	95	112	33	28	8					



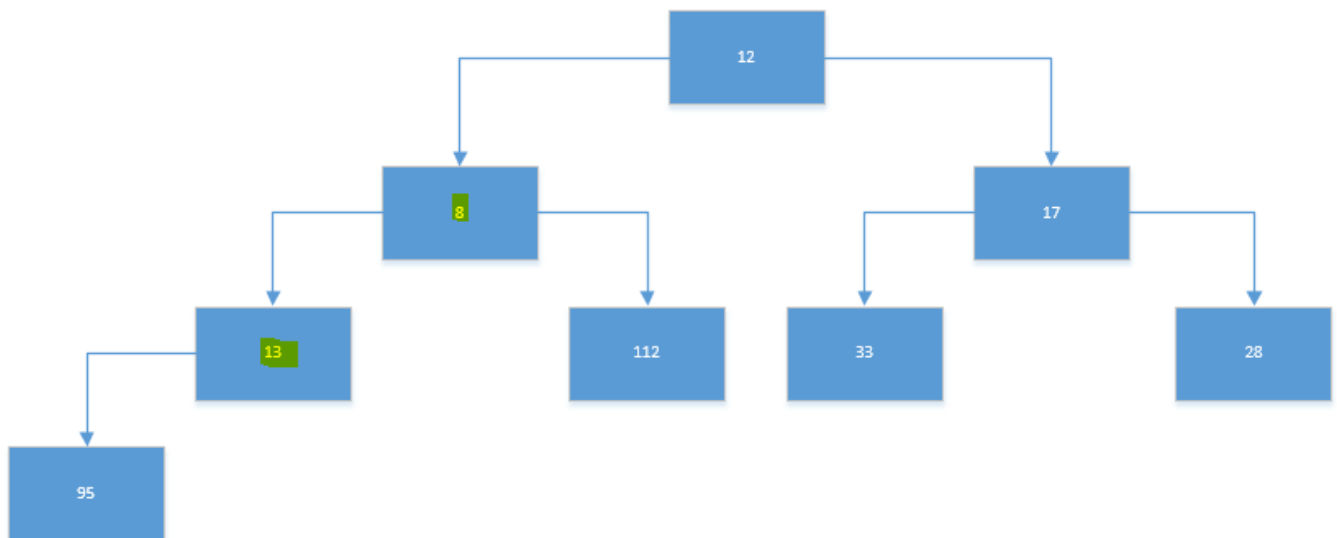
➤ $8 < 95 \rightarrow$ vertauschen

0	1	2	3	4	5	6	7	8	9	10	11	12	13
	12	13	17	8	112	33	28	95					



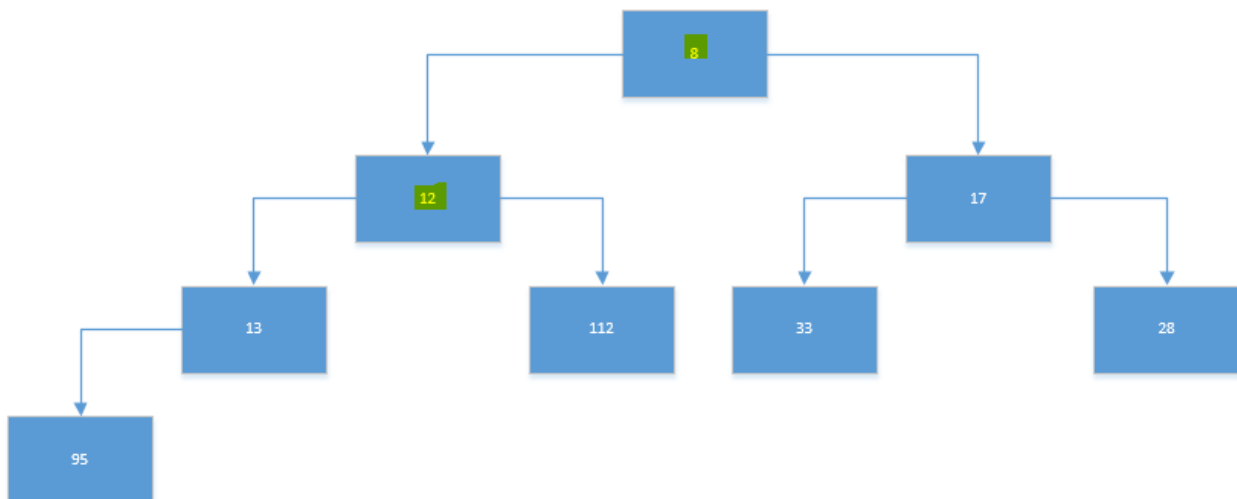
➤ $8 < 13 \rightarrow$ vertauschen

0	1	2	3	4	5	6	7	8	9	10	11	12	13
	12	8	17	13	112	33	28	95					



➤ $8 < 12 \rightarrow$ vertauschen

0	1	2	3	4	5	6	7	8	9	10	11	12	13
	8	12	17	13	112	33	28	95					

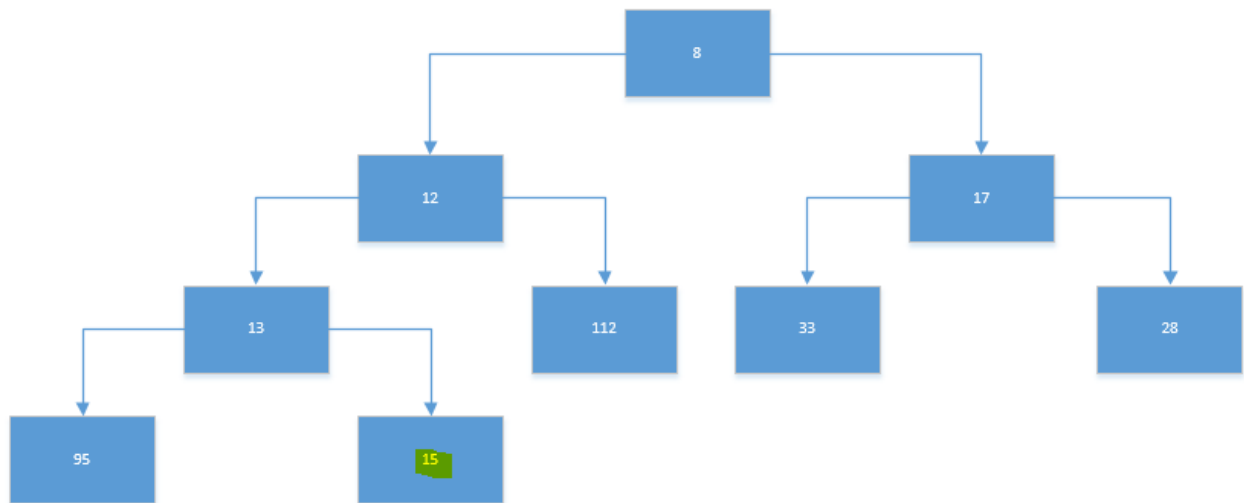


➤ $i = 1 \rightarrow$ Ende

Einfügen von 15:

➤ Einfügen am Ende

0	1	2	3	4	5	6	7	8	9	10	11	12	13
	8	12	17	13	112	33	28	95	15				

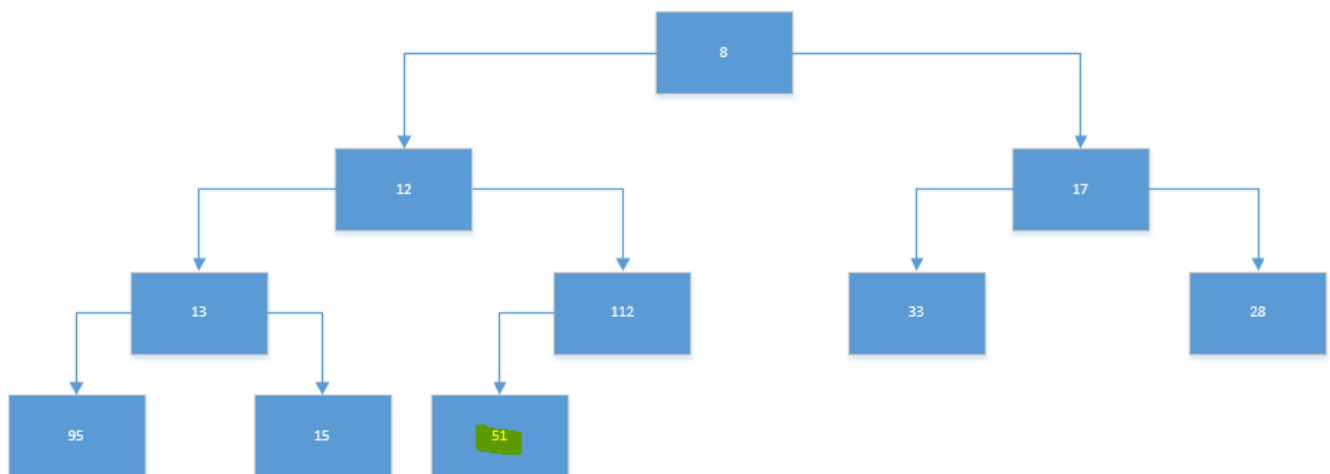


➤ $15 > 13 \rightarrow$ Ende

Einfügen von 51:

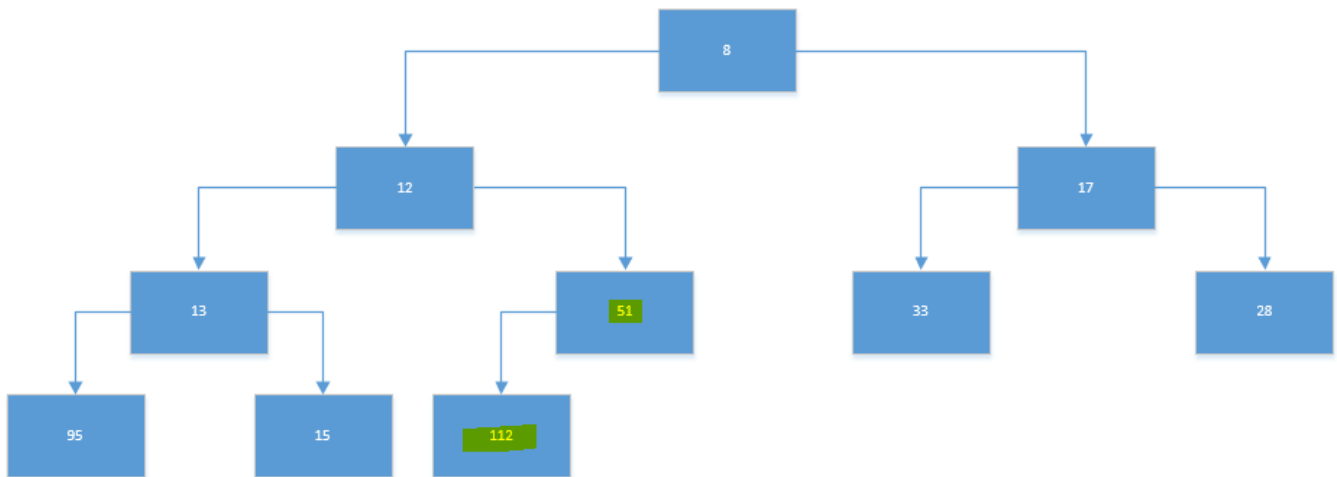
➤ Einfügen am Ende

0	1	2	3	4	5	6	7	8	9	10	11	12	13
	8	12	17	13	112	33	28	95	15	51			



➤ $51 < 112 \rightarrow$ vertauschen

0	1	2	3	4	5	6	7	8	9	10	11	12	13
	8	12	17	13	51	33	28	95	15	112			

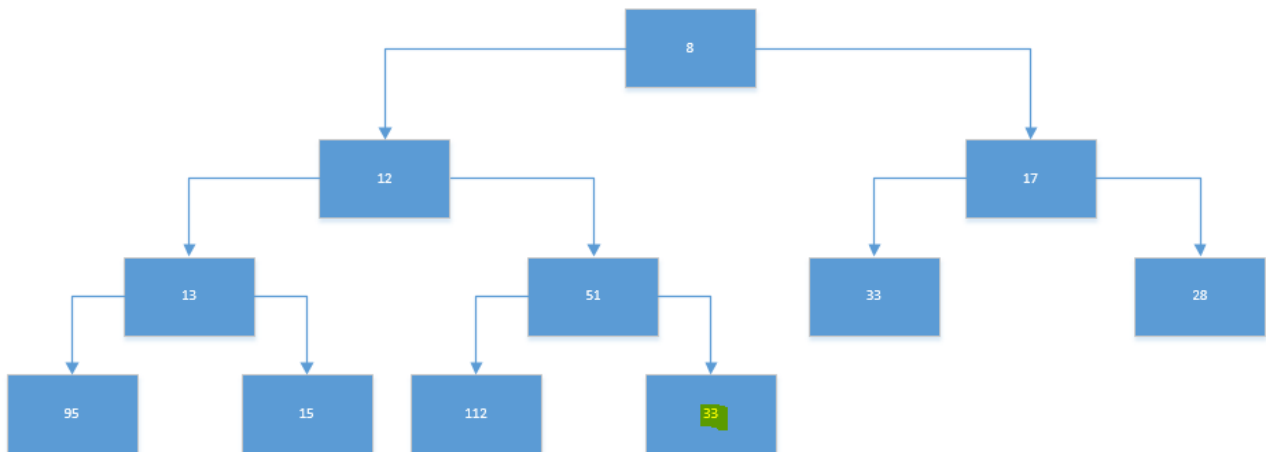


➤ $51 > 12 \rightarrow$ Ende

Einfügen von 33:

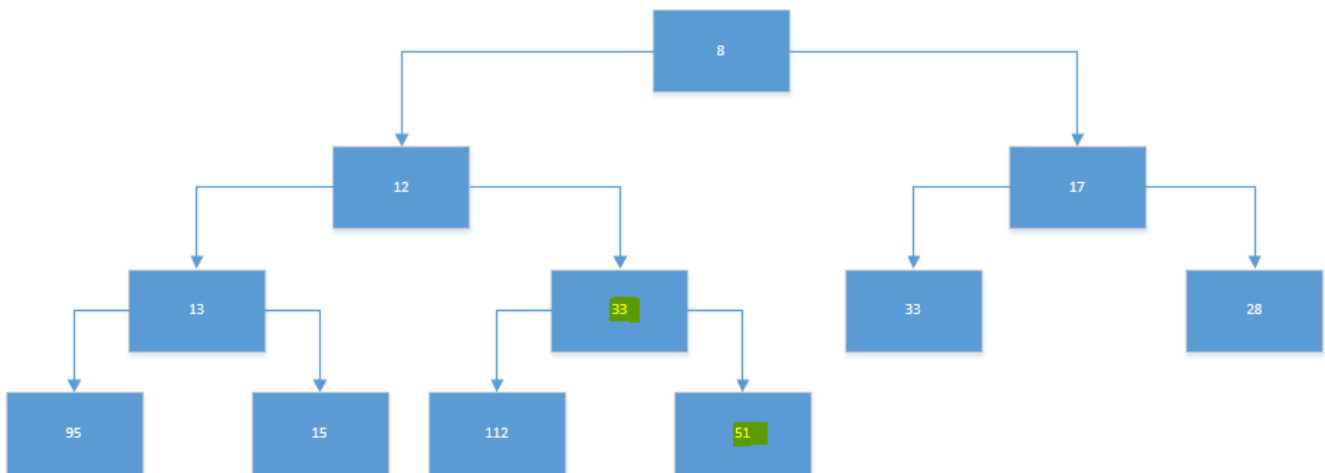
➤ Einfügen am Ende

0	1	2	3	4	5	6	7	8	9	10	11	12	13
	8	12	17	13	51	33	28	95	15	112	33		



➤ $33 < 51 \rightarrow$ vertauschen

0	1	2	3	4	5	6	7	8	9	10	11	12	13
	8	12	17	13	33	33	28	95	15	112	51		

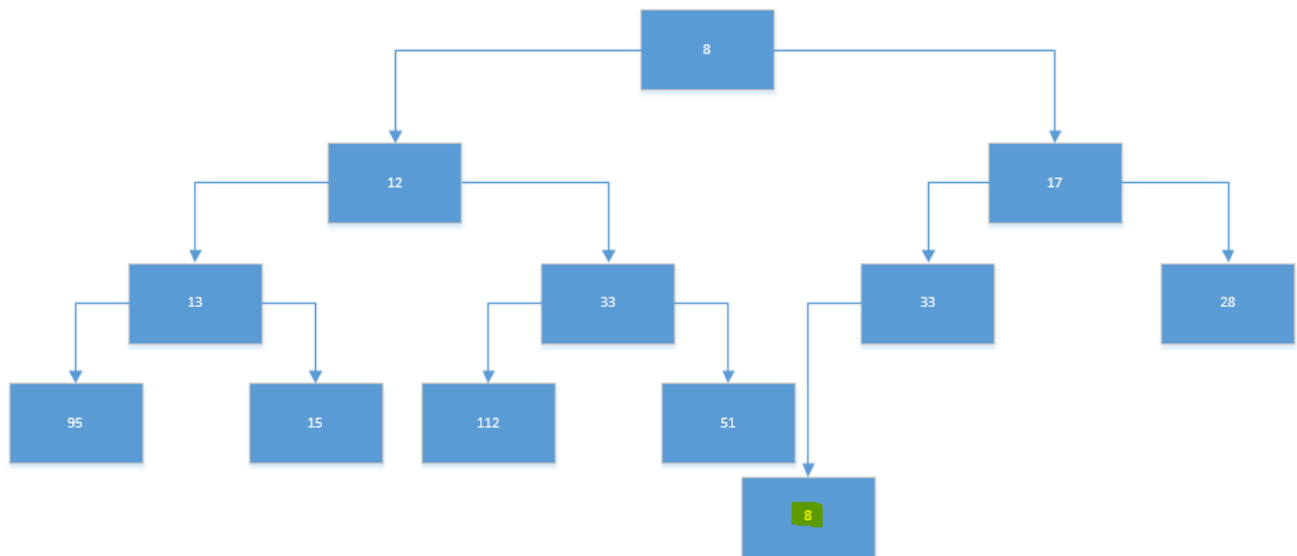


➤ $33 > 12 \rightarrow$ Ende

Einfügen von 8:

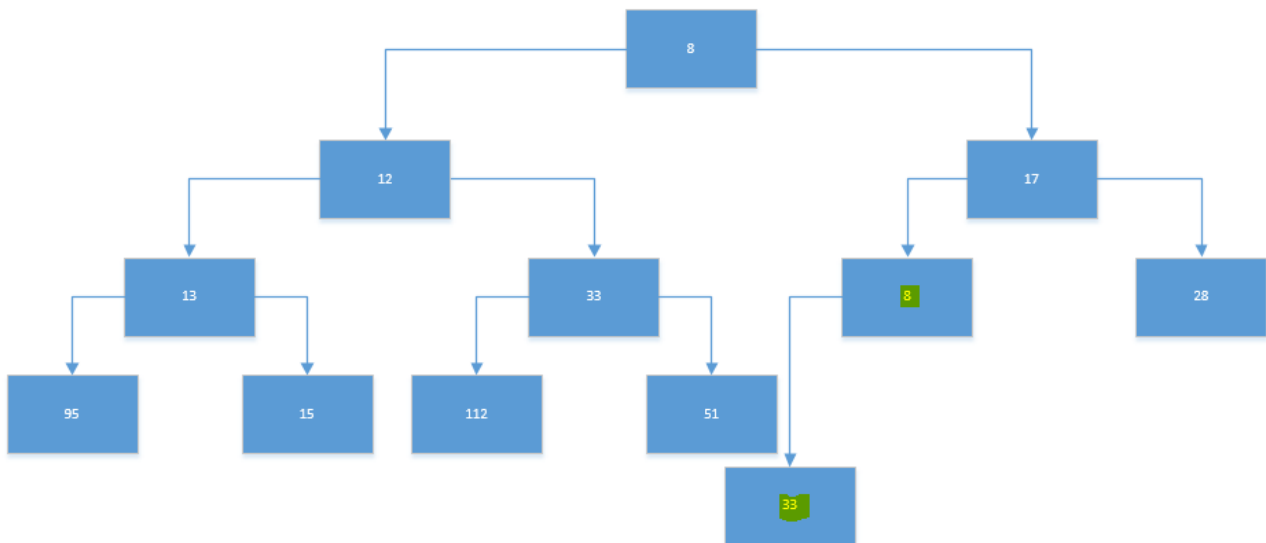
➤ Einfügen am Ende

0	1	2	3	4	5	6	7	8	9	10	11	12	13
	8	12	17	13	33	33	28	95	15	112	51	8	



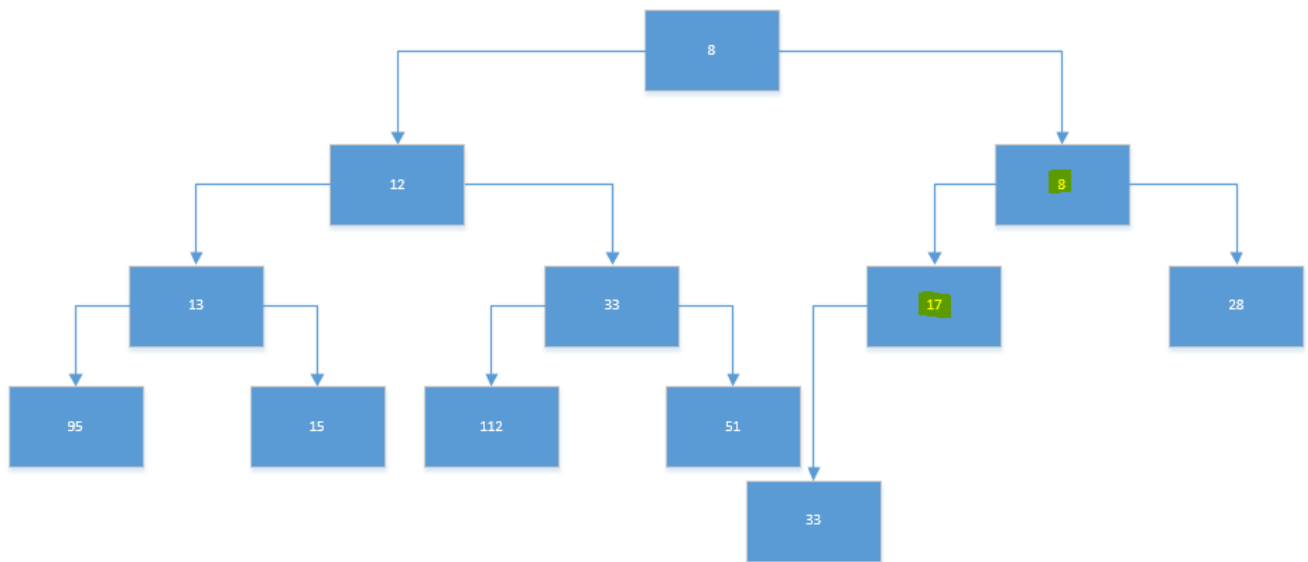
➤ $8 < 33 \rightarrow$ vertauschen

0	1	2	3	4	5	6	7	8	9	10	11	12	13
	8	12	17	13	33	8	28	95	15	112	51	33	



➤ $8 < 17 \rightarrow$ vertauschen

0	1	2	3	4	5	6	7	8	9	10	11	12	13
	8	12	8	13	33	17	28	95	15	112	51	33	



➤ 8 = 8 -> Ende