

Wurzel des B-Baums wird im Hauptspeicher behalten, um schnell darauf zuzugreifen. Knoten werden erst bei Bedarf in den Hauptspeicher geladen.

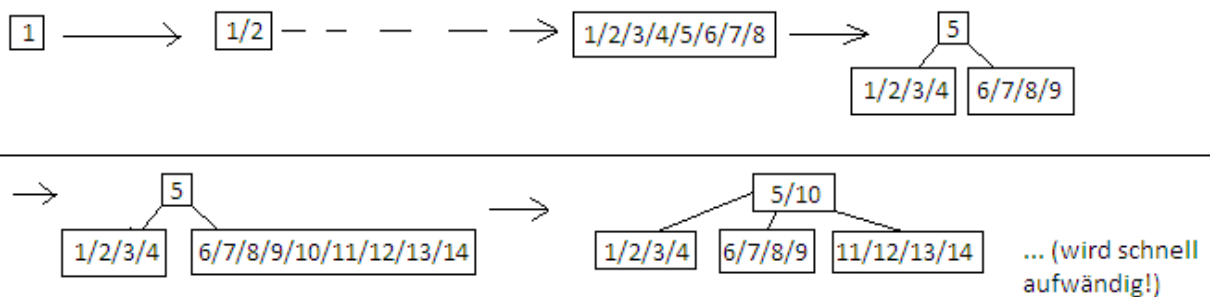
Wurzel hat mind. 1 Schlüssel, max. $2m$

Knoten hat mind. m , max. $2m$ Schlüssel

Knoten mit k Schlüsseln hat $k+1$ Söhne

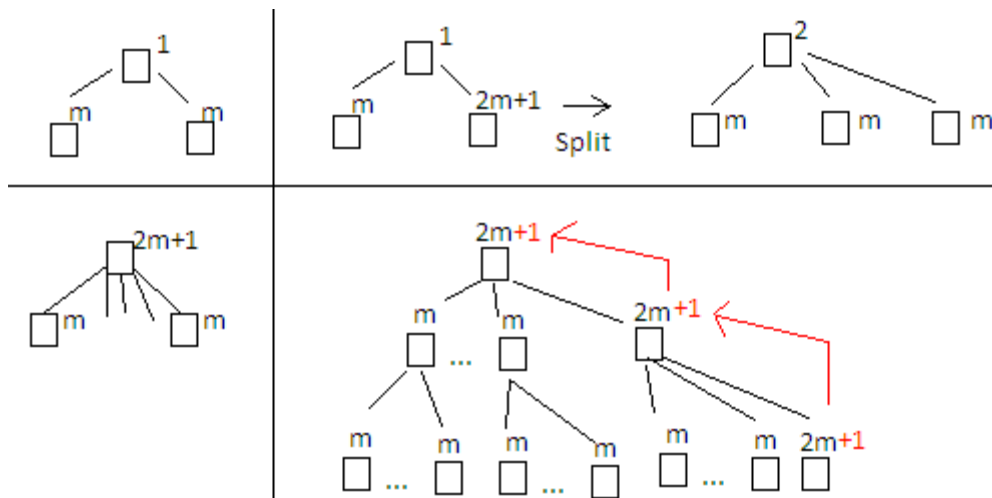
Aufgabe 4.1

$m=4$



Für den ersten Split: $2m+1$ Einfügungen (in den leeren Baum)

Dann $m+1$ Einfügungen, damit im Wurzelknoten ein Schlüssel mehr eingefügt wird



Vor dem Überlauf (also letztes Bild ohne rote Einfügung):

Maximal in Ebene 2: $2m$ Knoten mit je m Schlüsseln und ein Knoten mit $2m$ Schlüsseln:

$$2m * m + 2m$$

Maximal in Ebene 3:

$$2m * m * m * (m + 1) + (2m * m + 2m)$$

Hier:

$h = 2$: max. 48 Schlüssel = $(2m + 2mm + 2m)$

(Anzahl der Schlüssel in der Wurzel und der Ebene darunter, weil Höhe = 2)

=> Schlüssel 49 einfügen => Baum wächst

$h = 3$: max. 248 Schlüssel = $(2m + 2m + 2m + 2m * (m + 1) * m + 2m + 2m)$

=> Schlüssel 249 einfügen => Baum wächst

$$n \geq 2(m + 1)^{h-1} - 1 = 2 * (4 + 1)^{4-1} - 1 = 249$$

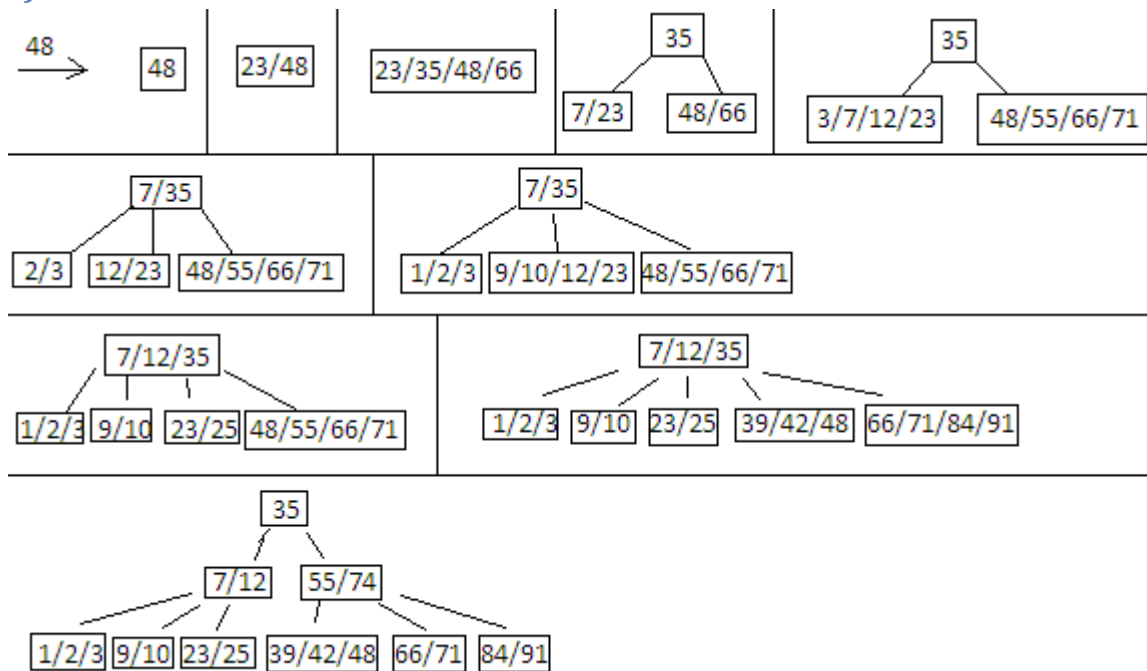
n = Anzahl der Schlüssel mit h = Höhe

=> Formel gibt an, mit welchen Schlüssel n man eine bestimmte Höhe h erreicht

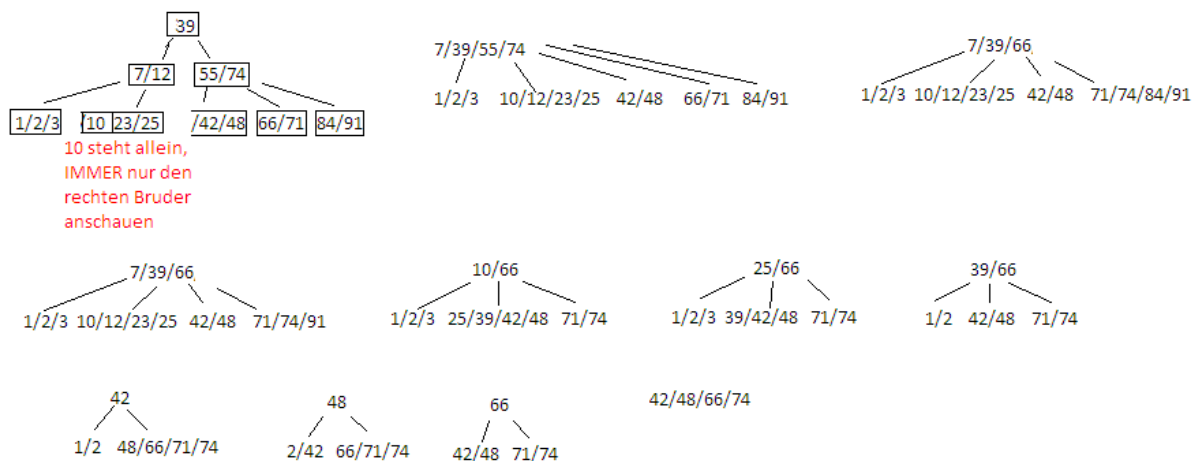
Aufgabe 4.2

$m=2$

a)



b)



Aufgabe 4.3

$m = 3$

B*-Baum: Knoten sind minimal mit $4/3 m$ gefüllt (Wurzel min. 1, max. $8/3 m$)

a)

