Exkurs: Binärbäume und B-Trees

Wie fängt ein Informatiker einen Löwen in Afrika?

(Anfänger Informatiker) Wir starten in Kairo und suchen ganz Afrika ab bis wir den Löwen gefunden haben (sehr, sehr aufwendig...)

-> Sequentielle Suche

(Schlauer Informatiker): Teile Afrika mit einem Zaun in zwei Teile ...

... prüfe in welcher Hälfte der Löwe ist ...

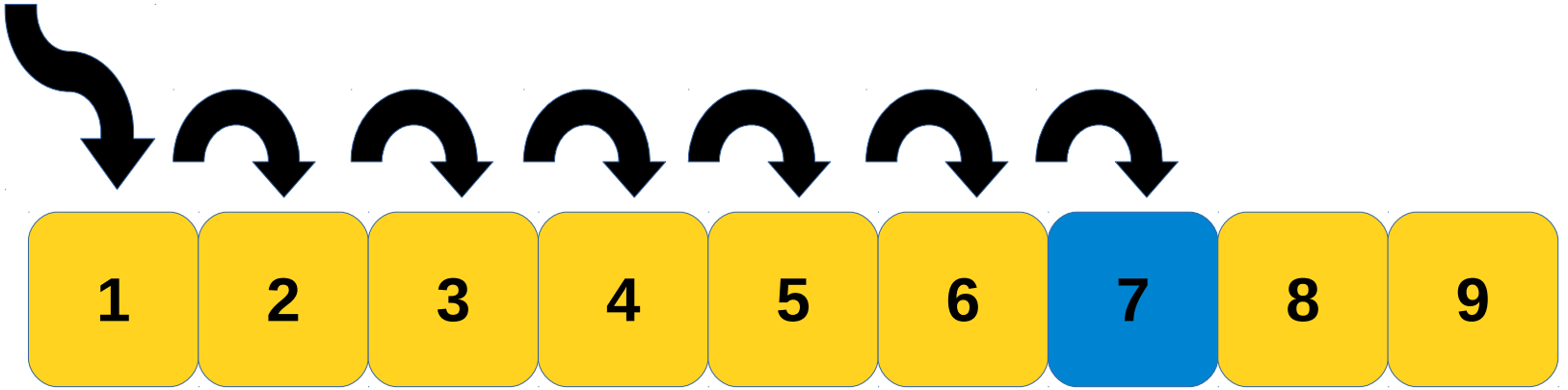
... teile die Hälfte wieder mit einem Zaun ...

... und mache so lange weiter ...

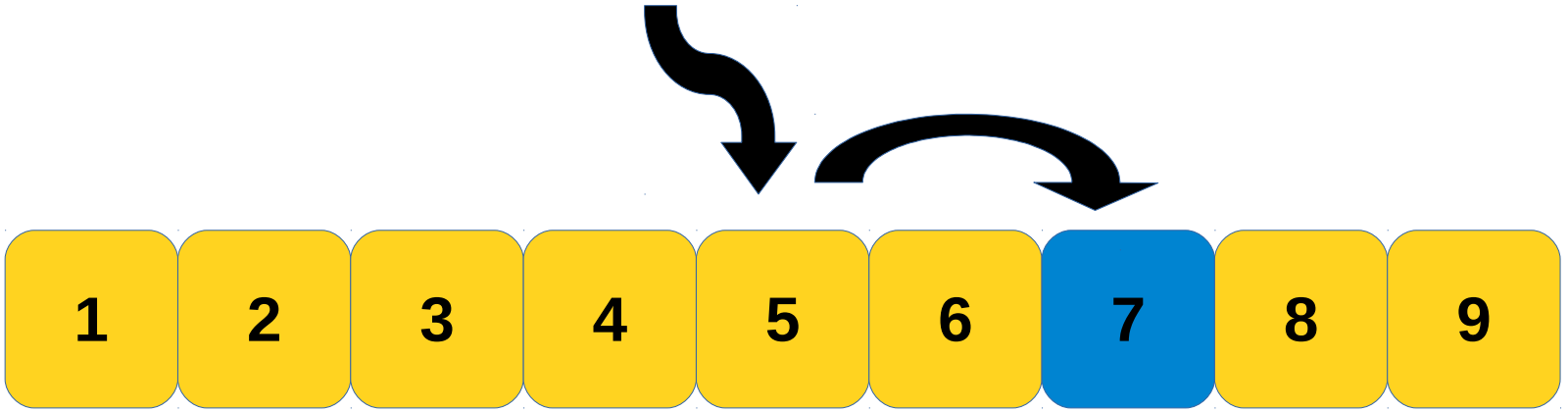
... bis wir den Löwen eingekreist haben!

-> Binärsuche

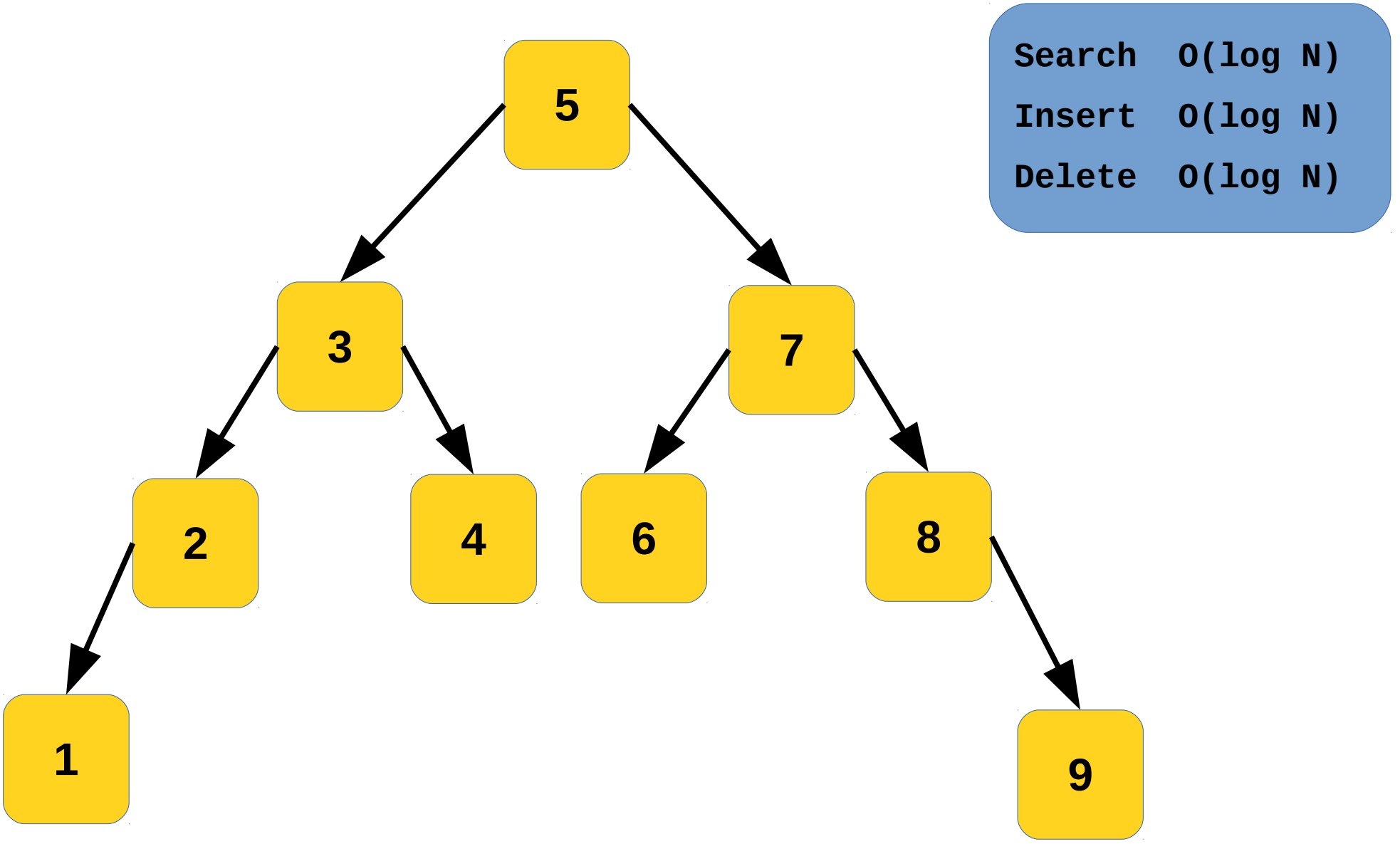
Sequentielle vs Binärsuche



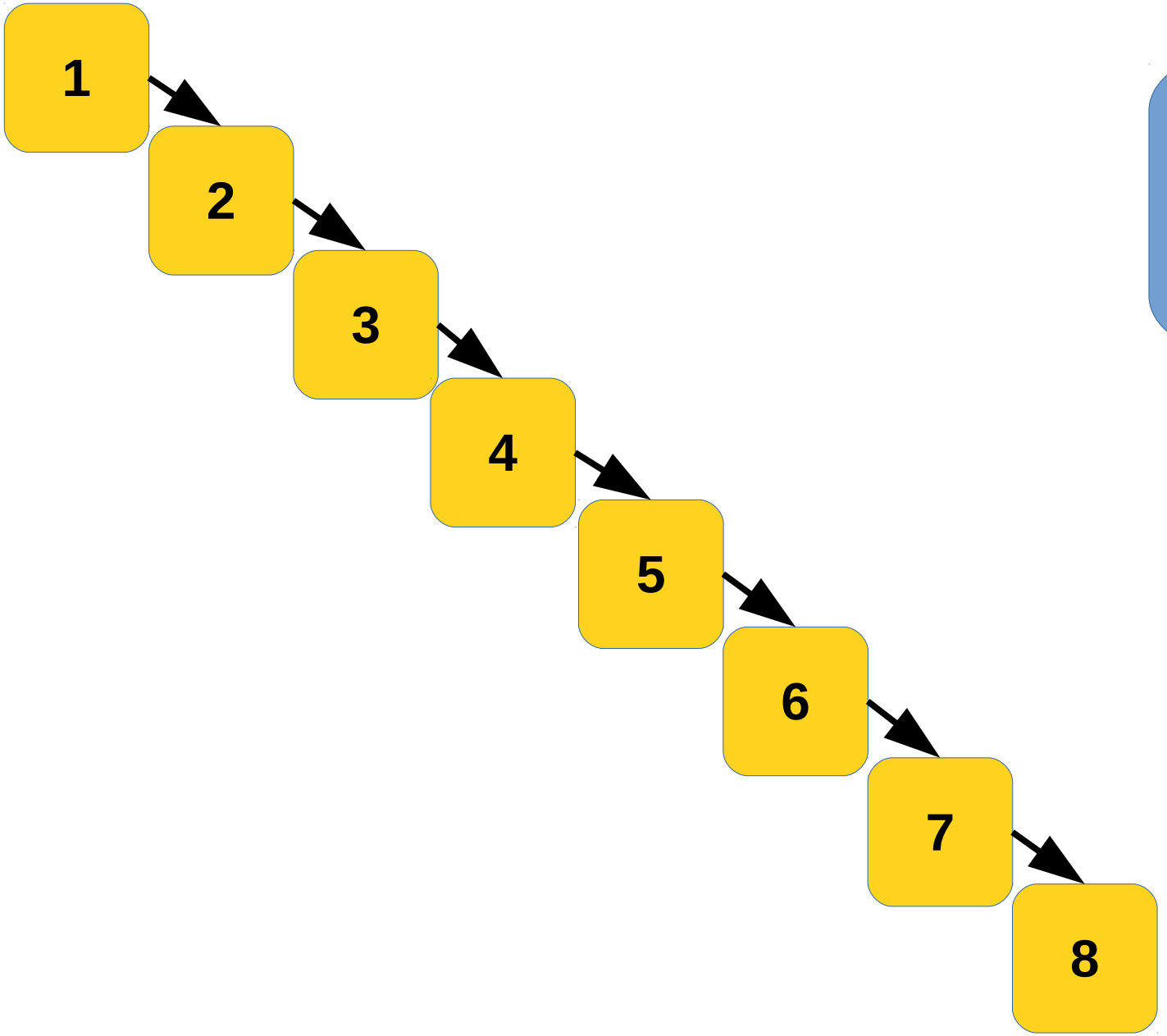

```
Get O(N)
Ins O(N)/O(1)
Del O(N)
```




```
Get O(log N)
Ins O(N)
Del O(N)
```


Search O(log N)
Insert O(log N)
Delete O(log N)
5
3
7
2
4
6
8
1
9

Binärbäume – Worst Case

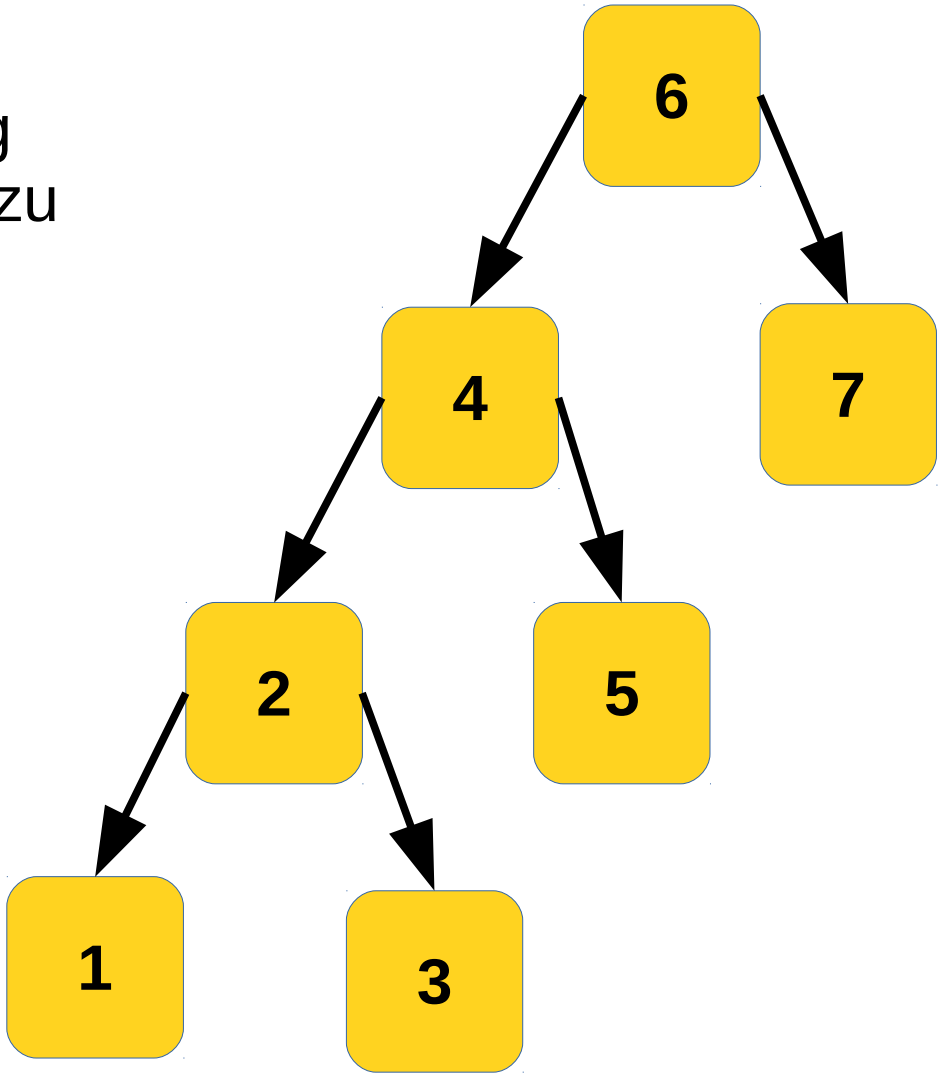


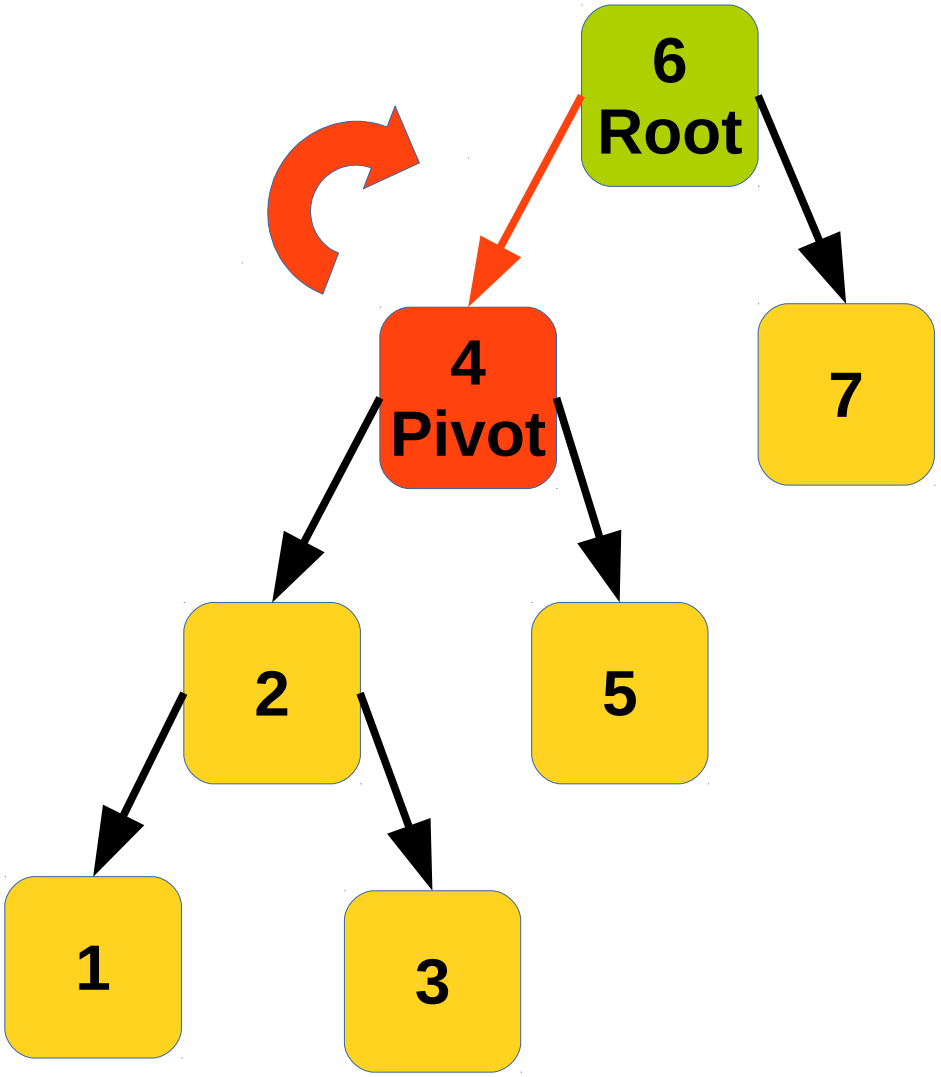
Search	O(N)
Insert	O(N)
Delete	O(N)

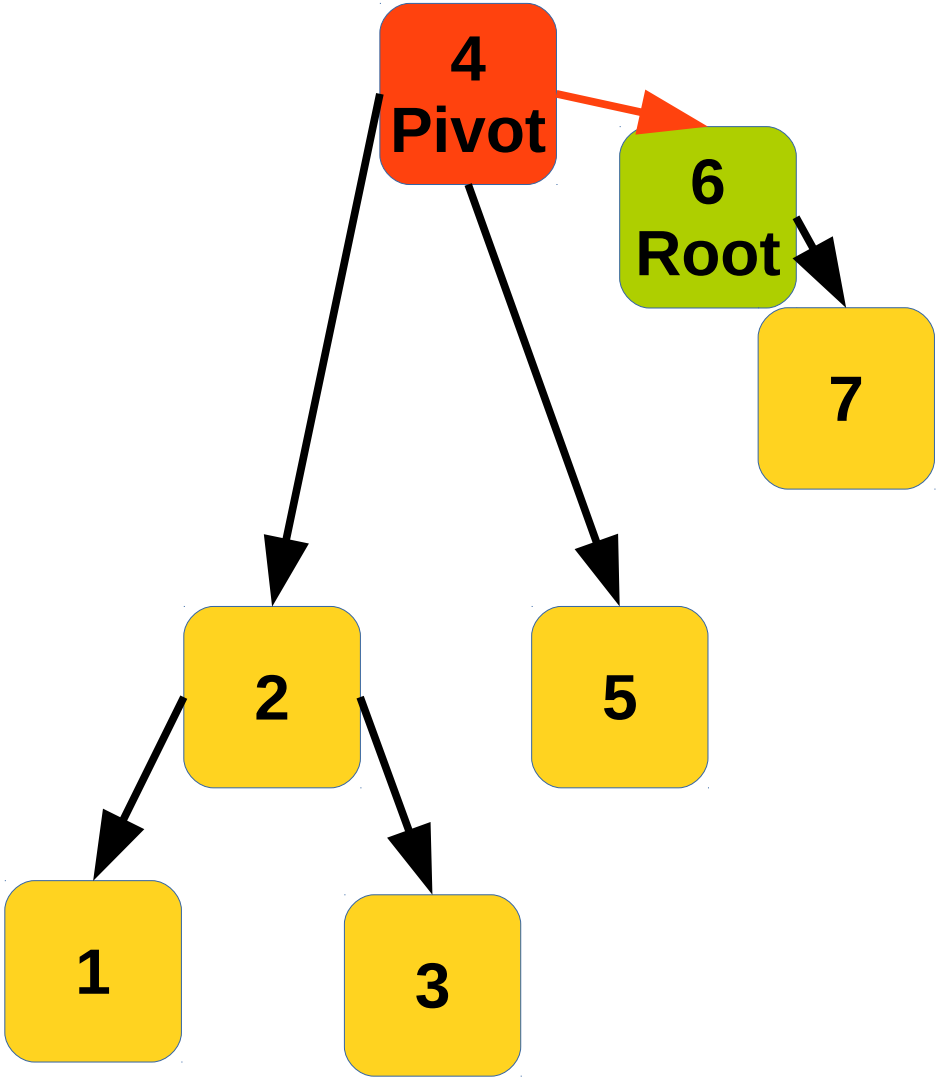
Balancing:

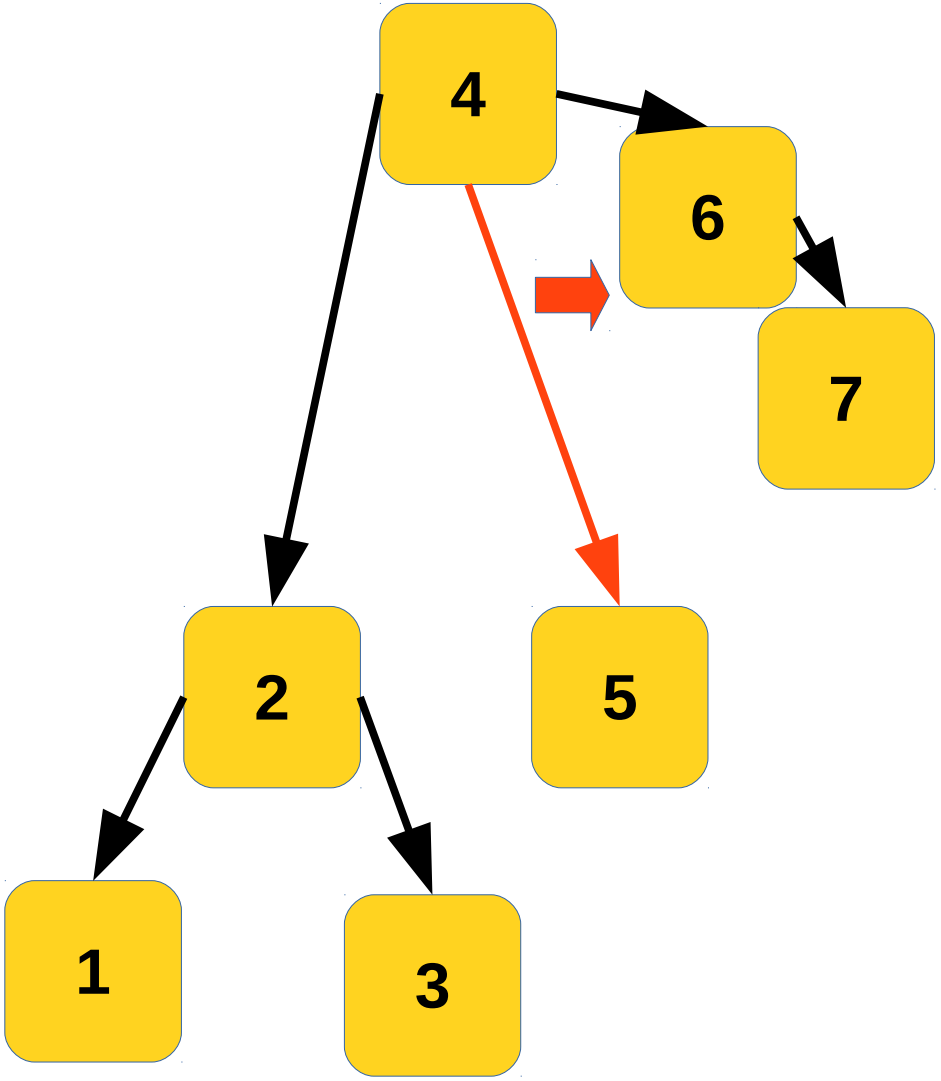
Nach jeder Änderung prüfen ob der Baum zu rechts / linkslastig wurde.

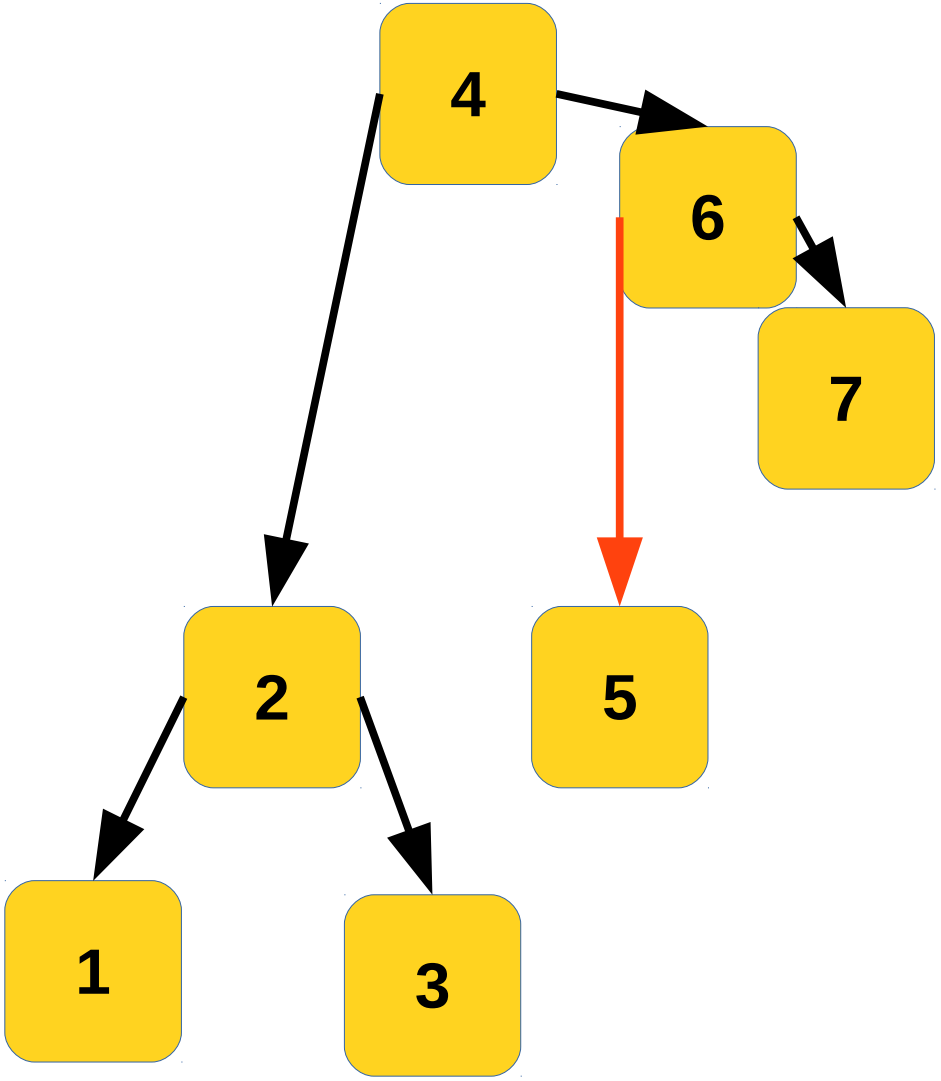
Falls nötig neu balancieren.



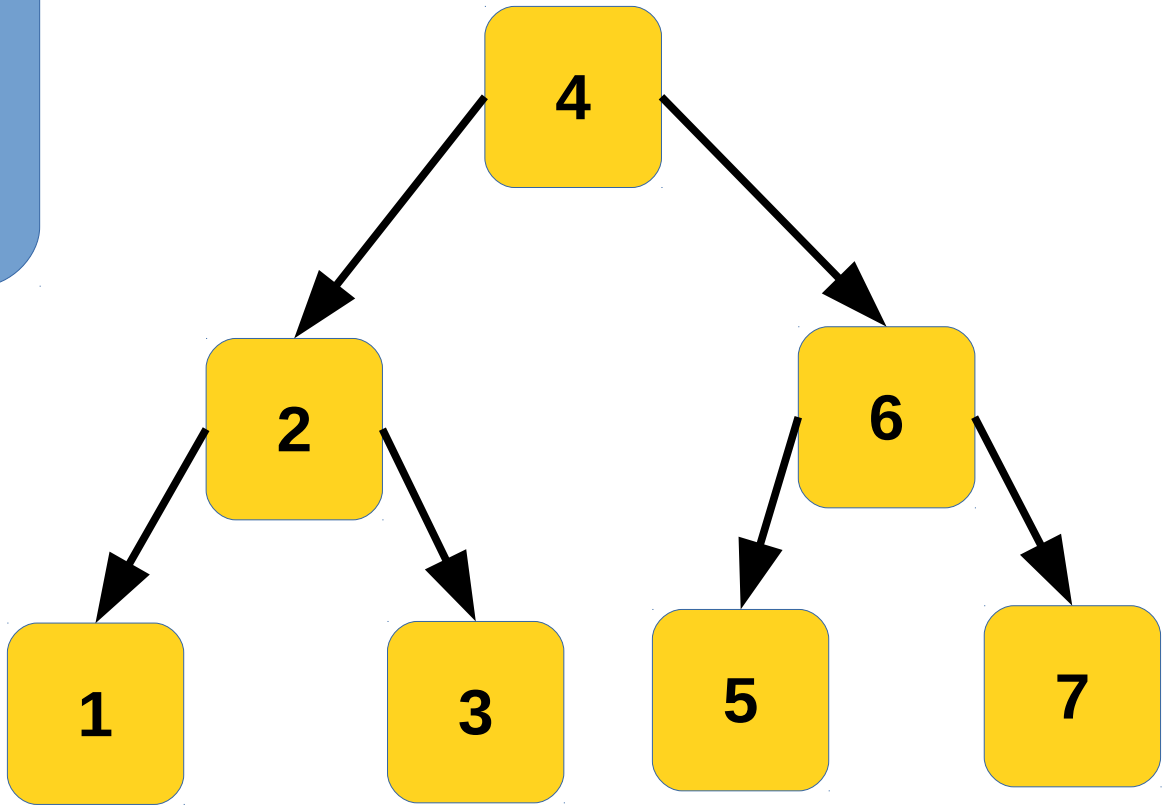

6
Root
4
Pivot
7
2
5
1
3


4
Pivot
6
Root
7
2
5
1
3


4
6
7
2
5
1
3


4
6
7
2
5
1
3

Worst-Case	
Search	O(log N)
Insert	O(log N)
Delete	O(log N)



- **AVL-Bäume**
  - Adelson-Velski / Landis, Sowjetunion 1962
  - Paper "An algorithm for the organization of information"

- **Red-Black Trees**
  - Knoten haben zusätzlich 'rote' oder 'schwarze' Farbe
  - Verbesserte Insert/Delete Performance

- **2-3 Trees**
  - Knoten können mehr als 2 Kinder haben
  - Performance ähnlich Red-Black

- **B-Trees**
  - Bayer / McCreight, Boeing Labs 1971
  - Knoten kann interne Knoten sowie variable Zahl an Kinderknoten haben
  - Weniger Rebalancing -> gut für Filesystem / Datenbanken!