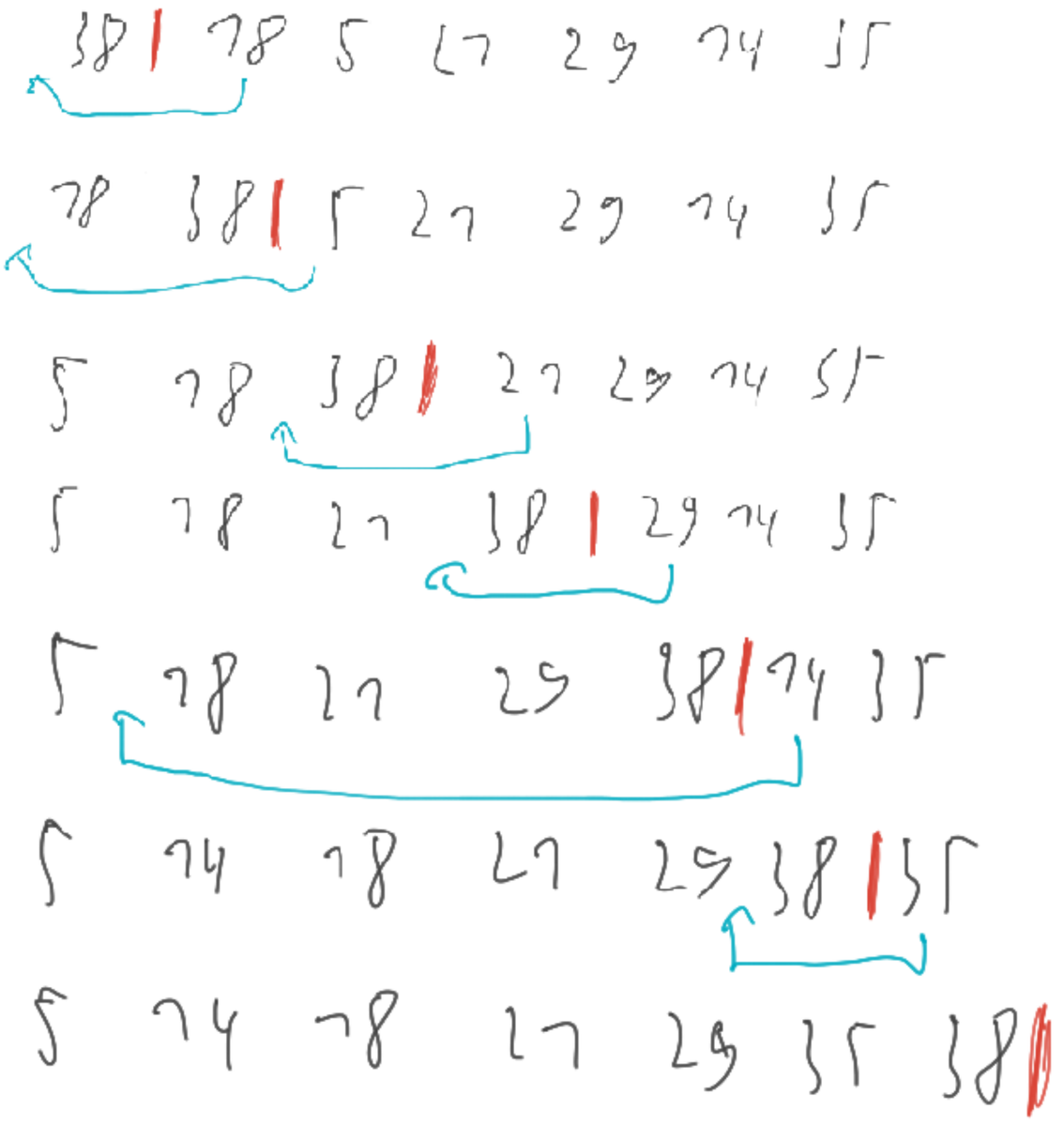
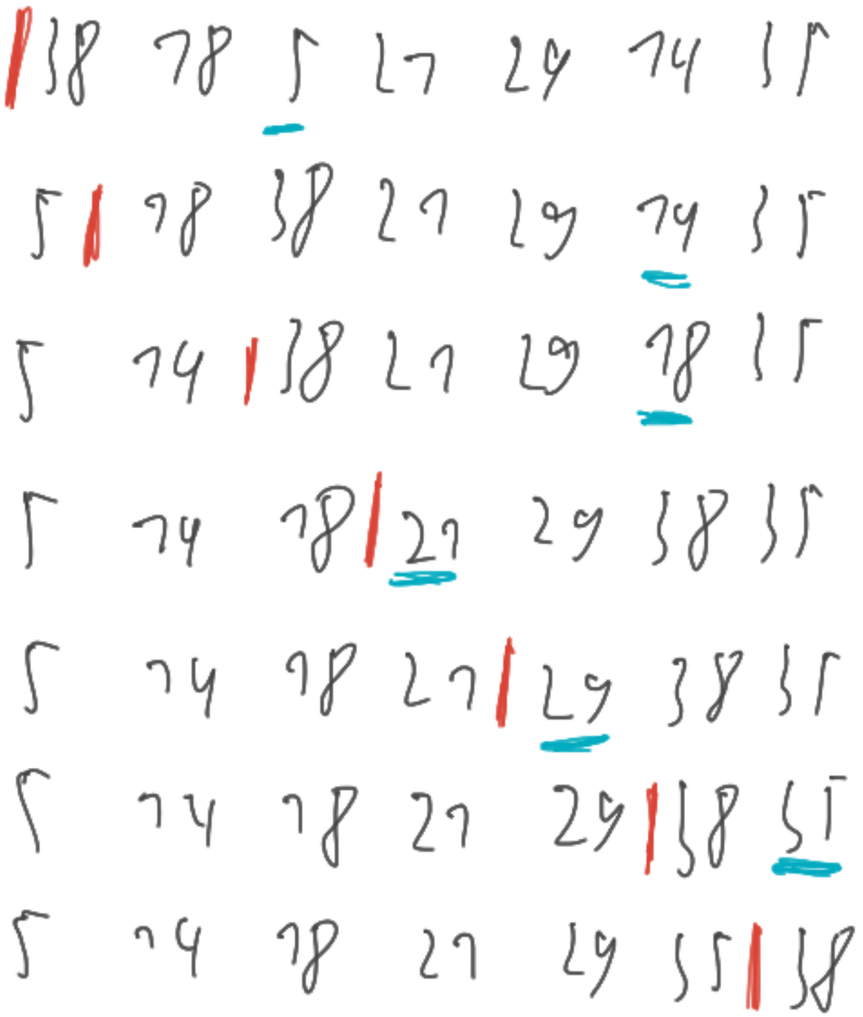
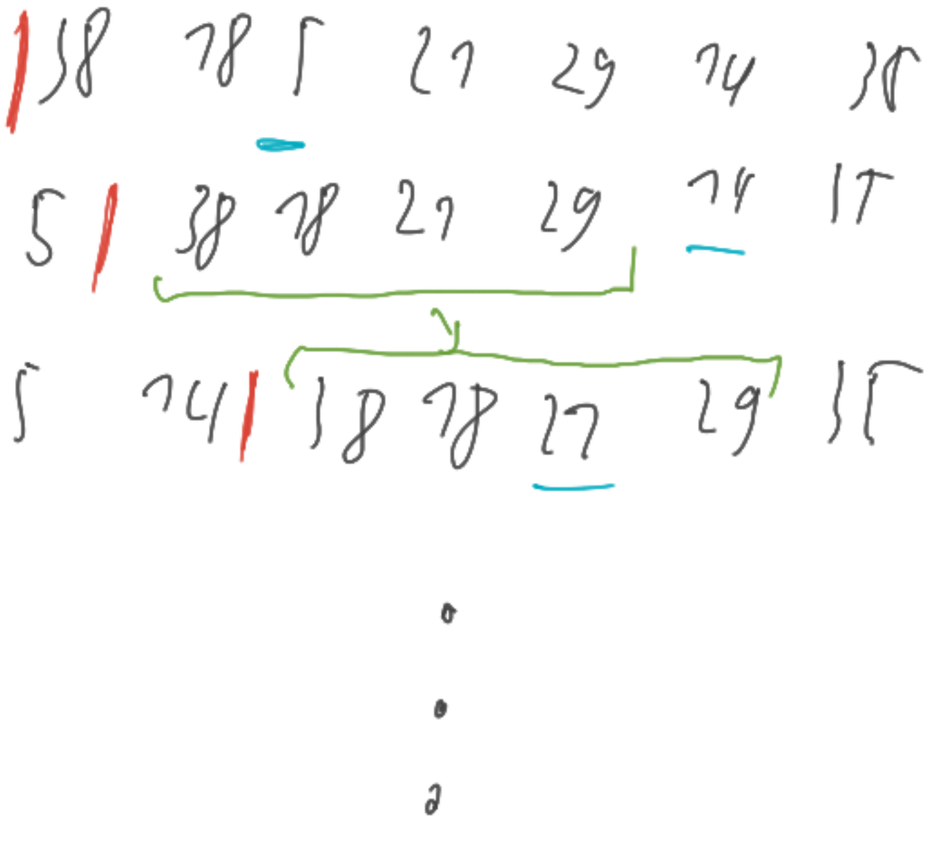
38 18 5 21 29 14 35



Insertion Sort

38 18 5 21 29 14 35





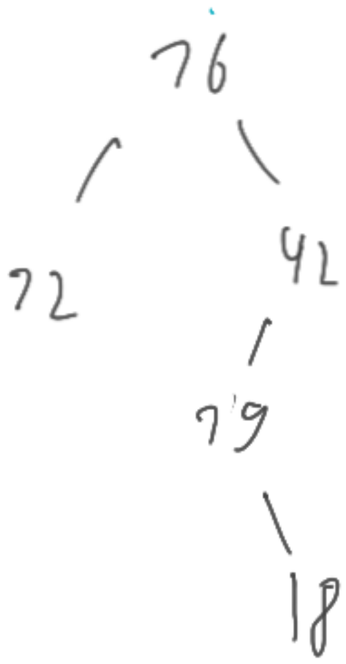
Selection Sort

AVL-Baum

42 16 12 19 38 1

AVL-Baum

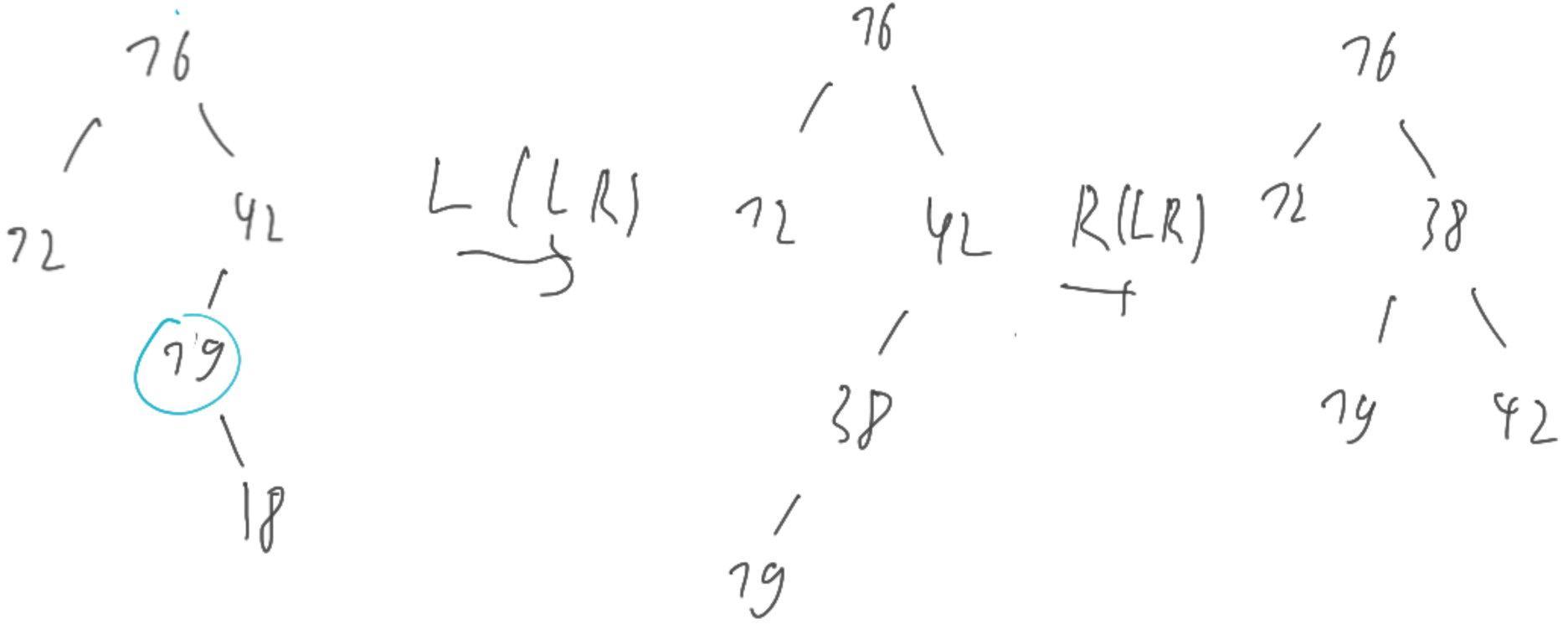
42 16 12 19 38 1



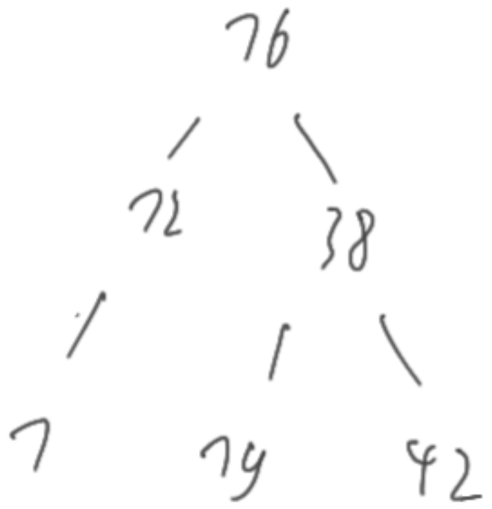
Einfache Rechts-Rotation reicht nicht!

42 16 12 19 38 1

AVL-Baum



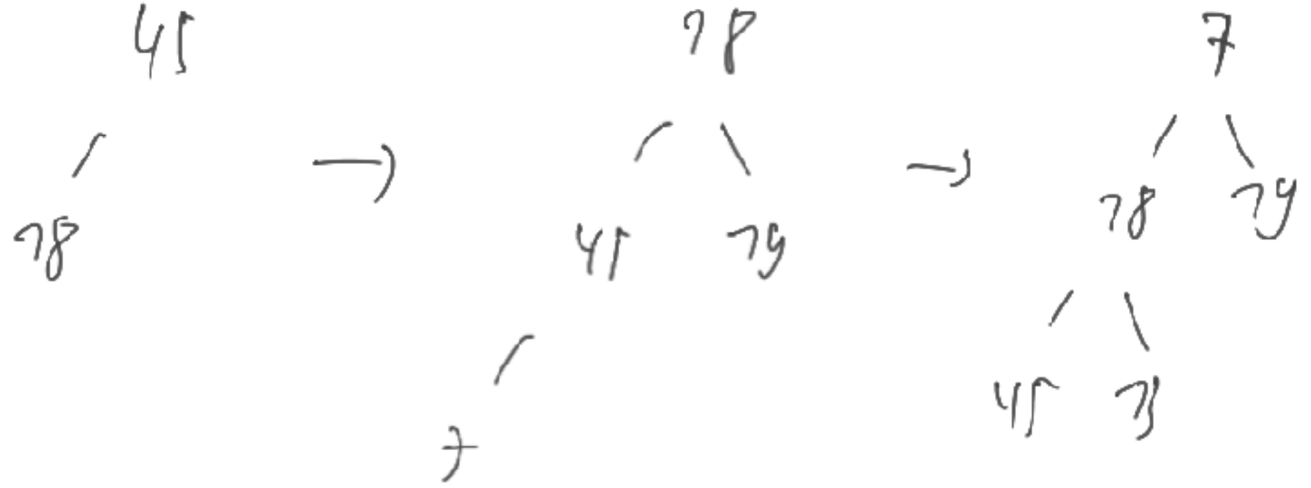
42 16 12 19 38 1

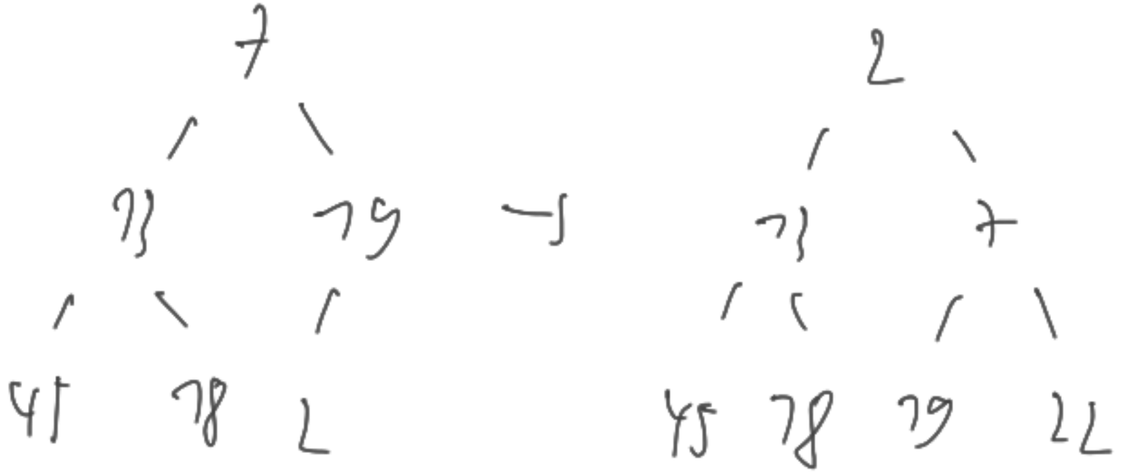


AVL-Baum

Min-Heap

45 18 19 7 13 2 22





FooSort

```
1   public static void foosort(List<Integer> list) {
2       if (list.size() == 0) {
3           return;
4       }
5       int max = list.get(0);
6       for (int i = 1; i < list.size(); i++) {
7           if (list.get(i) > max) {
8               max = list.get(i);
9           }
10      }
11      List<Integer> values = new ArrayList<>();
12      for (int i = 0; i <= max; i++) {
13          values.add(0);
14      }
15      for (int i = 0; i < list.size(); i++) {
16          values.set(list.get(i), values.get(list.get(i)) + 1);
17      }
18      int k = 0;
19      for (int i = 1; i < values.size(); i++) {
20          for (int j = 0; j < values.get(i); j++) {
21              list.set(k, i);
22              k++;
23          }
24      }
25  }
```

Sonderfall

Maximum suchen

Hilfsliste erzeugen

Hilfsliste befüllen

Werte zurückschreiben

Die Schleife hängt max+1 Nullen an die (anfangs leere) Liste an.

Hilfsliste

Speichert, wie oft jedes Element in list vorkommt.

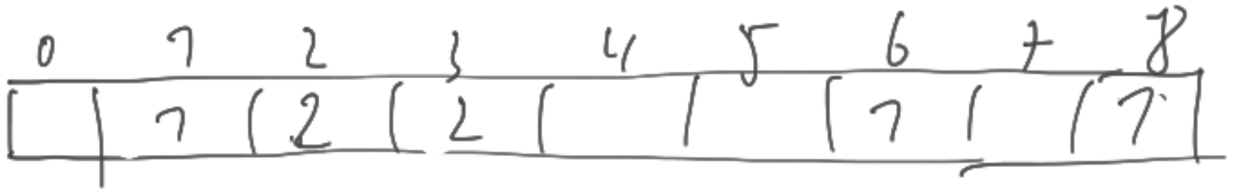
Hilfsliste hat eine Stelle für jeden möglichen Wert zwischen 0 und dem Maximum der Liste.

Hilfsliste enthält an Stelle i die Anzahl der Vorkommen von i in der ursprünglichen Liste

FooSort

list

values



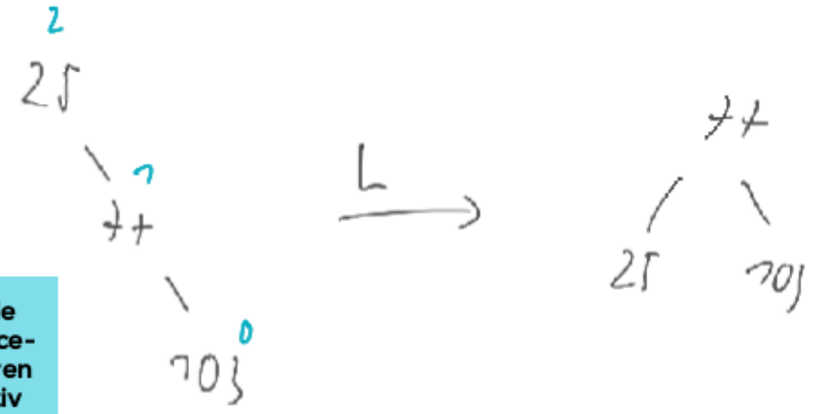
Lösung

Das Verfahren bestimmt zuerst das größte Element der Liste. Dann zählt es für jeden Wert der Liste, wie oft er vorkommt. Diese Häufigkeiten werden in einem Hilfsarray gespeichert.
Die sortierte Liste wird anschließend wieder aufgebaut, indem das Hilfsarray durchlaufen wird und dabei entsprechend jeder Häufigkeit die Elemente der Liste gesetzt werden.

Das Verfahren funktioniert nur für Listen,
die keine negativen Zahlen enthalten.

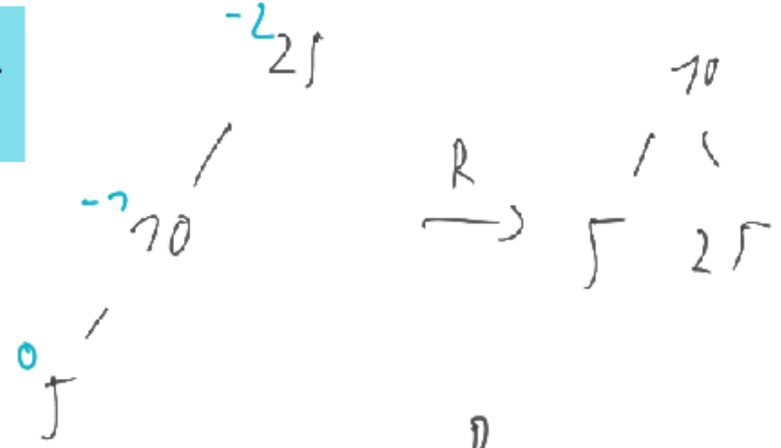
Kriterien für die Rotationen bei AVL-Bäumen

beide Balance-faktoren positiv



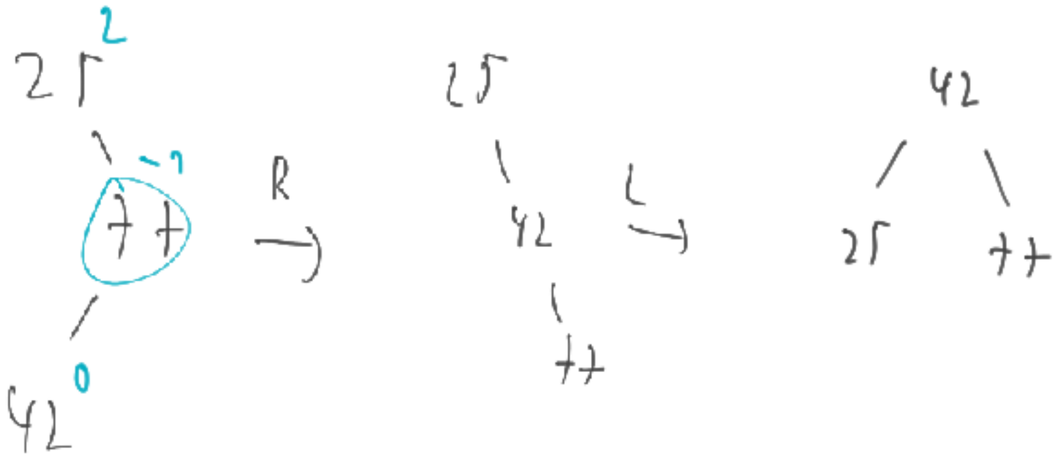
L

beide Balance-faktoren negativ

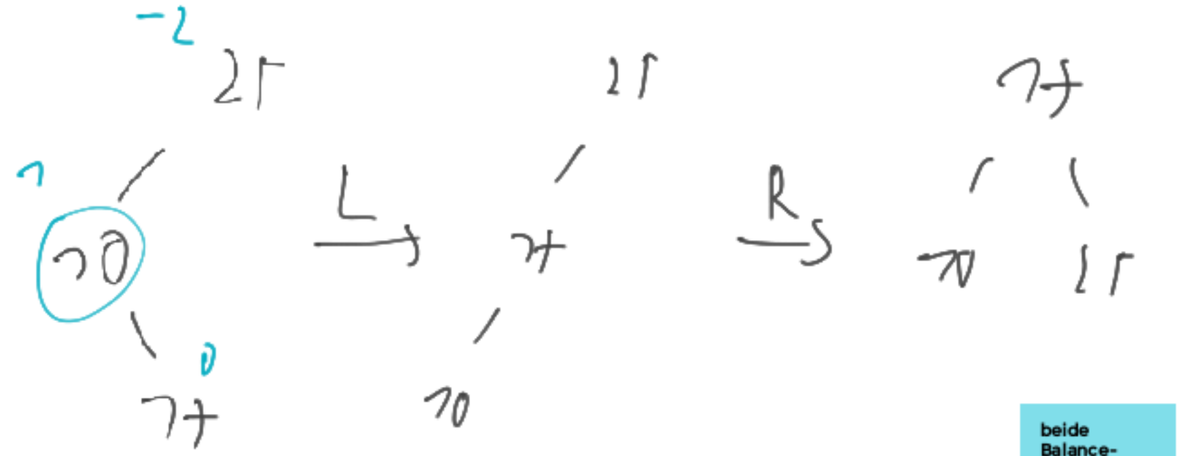


R

beide Balance-faktoren haben gleiches Vorzeichen



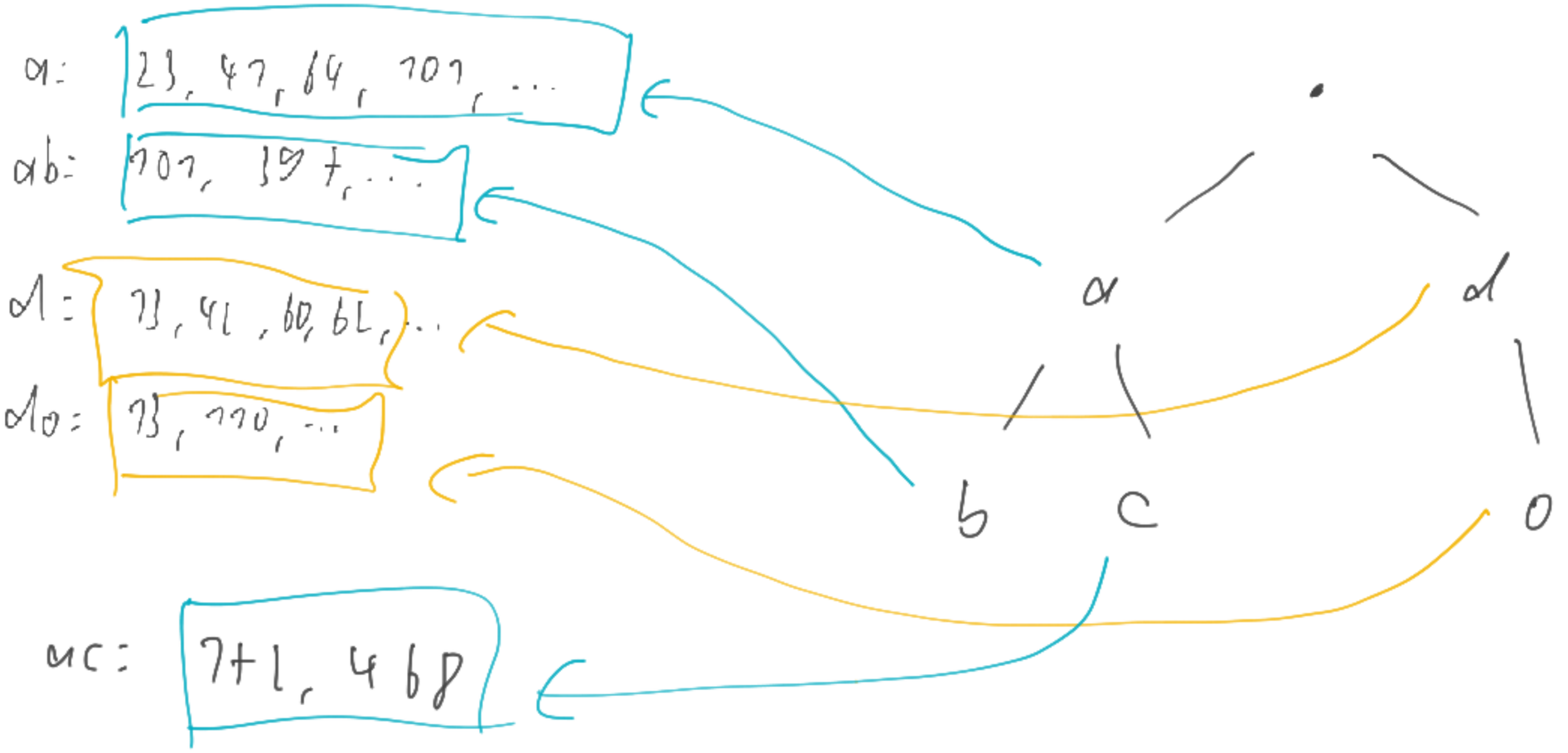
RL



L R

beide Balance-faktoren haben verschiedene Vorzeichen

Präfixbaum als Suchindex für einen Text



a Aa ab .* 1 von 46

```
# Lorem Ipsum

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore
magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd
gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur
sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam
voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata
sanctus est Lorem ipsum dolor sit amet.
```