

Frühjahr 2015 RS

1 Thema

Aufgabe 1

- i)
- ii)

Aufgabe 2

- i)
- ii)

Aufgabe 3

Aufgabe 4

Aufgabe 5

Aufgabe 6

- (a)
- (b)

Aufgabe 7

- (a) Zu zeigen: $f(n) \in O(g(n))$
 $g(n) \in O(f(n))$ bedeutet, dass $f(n)$ höchstens so schnell wächst wie $g(n)$.
 Somit gibt es eine Konstante c , so dass für große n immer gilt:

$$\exists c > 0 \exists x_0 > 0 \forall x > x_0 : |f(x)| \leq c \cdot |g(x)|$$

Sei $f(n) \in O(g(n))$

$$\Rightarrow \exists c > 0 : \exists n_0 > 0 : \forall n > n_0 : f(n) \leq c \cdot g(n)$$

$$\Rightarrow \exists c > 0 : \exists n_0 > 0 : \forall n > n_0 : 0,01n^3 + n - 1 \leq c \cdot (100(n+2)^2 + 30)$$

$$\Rightarrow \exists c > 0 : \exists n_0 > 0 : \forall n > n_0 : 0,01n^3 + n - 1 \leq c \cdot (100n^2 + 400n + 430)$$

$$\Rightarrow \exists c > 0 : \exists n_0 > 0 : \forall n > n_0 : \frac{0,01n^3 + n - 1}{100n^2 + 400n + 430} \leq c \quad \Big| \lim_{n \rightarrow \infty}$$

$$\Rightarrow \exists c > 0 : \exists n_0 > 0 : \forall n > n_0 : \lim_{n \rightarrow \infty} \frac{0,01n^3 + n - 1}{100n^2 + 400n + 430} \leq c$$

$$\Rightarrow \exists c > 0 : \exists n_0 > 0 : \forall n > n_0 : \infty \leq c \quad \text{!}$$

$$\Rightarrow f(n) \notin O(g(n))$$

- (b) Zu zeigen: $g(n) \in O(f(n))$
 $g(n) \in O(f(n))$ bedeutet, dass $g(n)$ höchstens so schnell wächst wie $f(n)$.
 Somit gibt es eine Konstante c , so dass für große n immer gilt:

$$\exists c > 0 \exists x_0 > 0 \forall n > n_0 : |g(n)| \leq c \cdot |f(n)|$$

Sei $c = 90.000$

$$\forall n > 1 : 100n^2 + (400 - 90.000)n + (430 + 90.000) \leq 900n^3 + 0n + 0$$

$$\Rightarrow \exists c > 0 : \exists n_0 > 0 : \forall n > n_0 : 100n^2 + (400 - c)n + (430 + c) \leq 0,01n^3 \cdot c$$

$$\Rightarrow \exists c > 0 : \exists n_0 > 0 : \forall n > n_0 : 100n^2 + 400n - cn + 430 + c \leq 0,01n^3 \cdot c$$

$$\Rightarrow \exists c > 0 : \exists n_0 > 0 : \forall n > n_0 : 100n^2 + 400n + 430 \leq 0,01n^3 \cdot c + n \cdot c - 1 \cdot c$$

$$\Rightarrow \exists c > 0 : \exists n_0 > 0 : \forall n > n_0 : 100(n+2)^2 + 30 \leq c \cdot (0,01n^3 + n - 1)$$

$$\Rightarrow \exists c > 0 : \exists n_0 > 0 : \forall n > n_0 : g(n) \leq c \cdot f(n)$$

$$\Rightarrow g(n) \in O(f(n))$$

Aufgabe 8

6	3	17	2	1	20	4	9	12	5
3	2	1	4	5	6	17	20	9	12
2	1	3	4	5	6	9	12	17	20
1	2	3	4	5	6	9	12	17	20
1	2	3	4	5	6	9	12	17	20

In jedem Rekursionsaufruf wird für den jeweiligen Abschnitt der Reihung ein Pivotelement gewählt (gelbe Einfärbung).

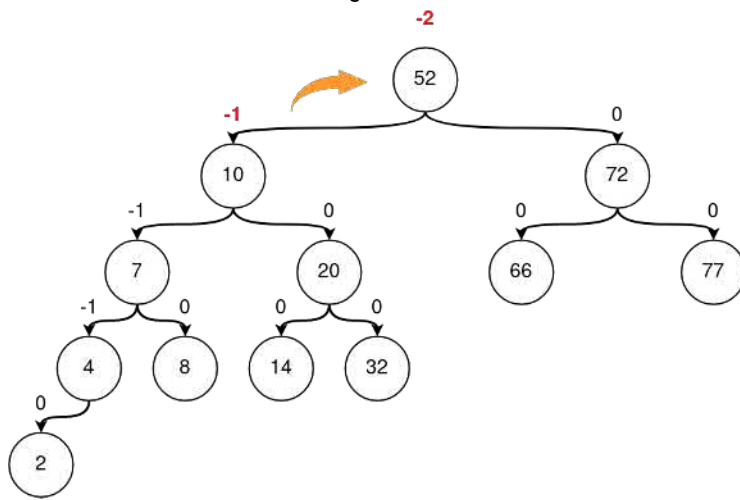
Das Sortierverfahren geht dann die Elemente der Reihung durch und platziert Elemente, die kleiner als das Pivotelement sind, links von diesem, und zu seiner Rechten die, die größer sind. Das Pivotelement steht somit im Anschluss an den Durchlauf an sortierter Stelle und braucht nicht weiter beachtet werden. Dies soll hier mit grüner Einfärbung markiert werden. Die Rekursion setzt sich kaskadenartig fort, deshalb sind je Durchlauf ab Schnitt 2 entsprechend mehrere Pivotelemente markiert, das gelb markierte Element ist dann das Pivotelement für den jeweils durchgereichten Abschnitt der Reihung.

Sie bricht erst dann ab, wenn nur noch ein einzelnes Element als Eingabe erhält, deshalb müssen die noch verbleibenden Einzelelemente in Schritt 4 nochmal einzeln rekursiv verarbeitet werden, obwohl sie eigentlich schon an sortierter Stelle stehen. In Schritt 5 gibt es keine unverarbeiteten Elemente mehr. Damit endet die Rekursion und die Reihung ist sortiert.

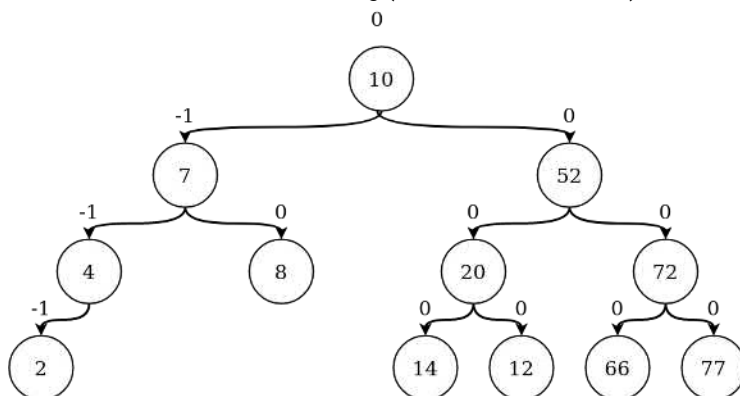
Aufgabe 9

- (a) Der Baum verletzt die Höhenbalance-Eigenschaft bei Knoten 5.
- (b) Durch das Einfügen wird die Höhenbalance-Eigenschaft an der Position von Knoten 52 verletzt. Dessen Balancefaktor sowie der des linken Kindknotens sind beide negativ; somit ist eine einfache Rechtsrotation erforderlich:

Der Baum vor der Rebalancierung:



Der Baum nach der Rebalancierung (mittels Rechtsrotation):



Aufgabe 10

a	b	c	d	e	f	g	h	Prioritätswarteschlange
0	∞	∞	∞	∞	∞	∞	∞	a
-	7	∞	9	10	12	∞	3	h, b, d, e, f
-	7	∞	8	9	5	12	-	f, b, d, e, g
-	7	9	8	6	-	11	-	e, b, d, c, g
-	7	9	8	-	-	8	-	b, d, g, c
-	-	9	8	-	-	8	-	d, g, c
-	-	9	-	-	-	8	-	g, c
-	-	9	-	-	-	-	-	g
-	-	-	-	-	-	-	-	-
0	7	9	8	6	5	8	3	

2 Thema

Aufgabe 1

- 1.
- 2.

Aufgabe 2

- 1.
- 2.
- 3.

Aufgabe 3

(a)

0	1	2	3	4
B	A			
D	C			
	E			

(b)

0	1	2	3	4
B	A	D	C	E

(c)

```

1  void insert(String s, int hs) {
2      assert s != null && 0 <= hs && hs <
        buckets.length;
3      if (buckets[hs] == null)
4          buckets[hs] = new LinkedList<>();
5      if (!buckets[hs].contains(s))
6          buckets[hs].addLast(s);
7  }
```

Aufgabe 4

(a)

```
1  void insert(String s) {
2      assert s != null;
3      for (char c : s.toCharArray())
4          assert c >= 'A' && c <= 'Z';
5      Trie curr = this;
6      for (char c : s.toCharArray()) {
7          if (curr.children[c - 'A'] == null)
8              curr.children[c - 'A'] = new Trie();
9          curr = curr.children[c - 'A'];
10     }
11 }
```

(b) $O(n)$

(c)

```
1  void printSorted() {
2      if (myString != null)
3          System.out.println(myString + " ");
4      for (Trie t : children)
5          if (t != null)
6              t.printSorted();
7  }
```

(d)

```
1  boolean contains(String s) {
2      if (s == myString)
3          return true;
4      for (Trie t : children)
5          if (t.contains(s))
6              return true;
7      return false;
8  }
```

Aufgabe 5

(a)

(b)