

Bäume Kurztest Besprechung

Algorithmen und Datenstrukturen 2

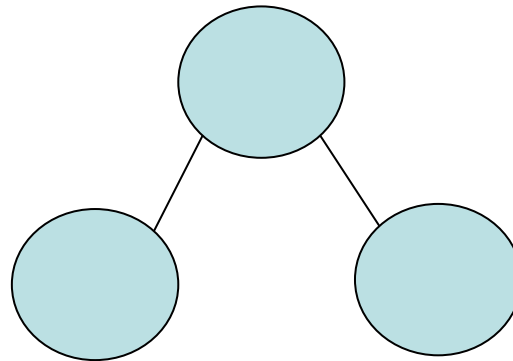
- Grüne Farbe: Bitte im Script nachtragen

Aufgabe a.) Achtung: Set-Semantik (deshalb LinkedList add in $O(n)$)

Operation	Linked List	Binärer Suchbaum	AVL Baum
add(element)	$O(n)$	$O(n)$	$O(\log n)$
contains(object)	$O(n)$	$O(n)$	$O(\log n)$
remove(element)	$O(n)$	$O(n)$	$O(\log n)$

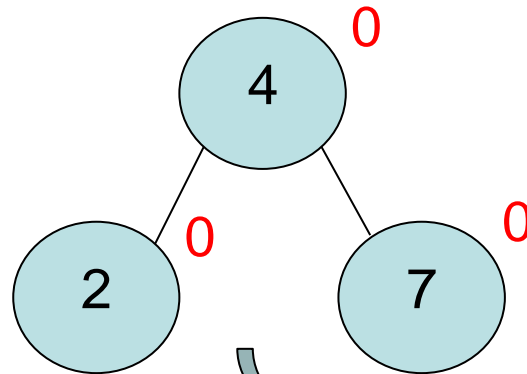
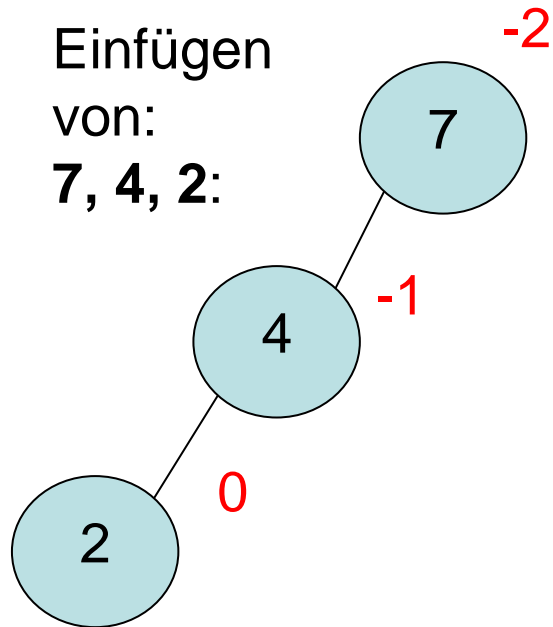
Aufgabe b.)

- Es gibt nur eine gültige Anordnung für 3 Knoten in einem AVL-Baum:

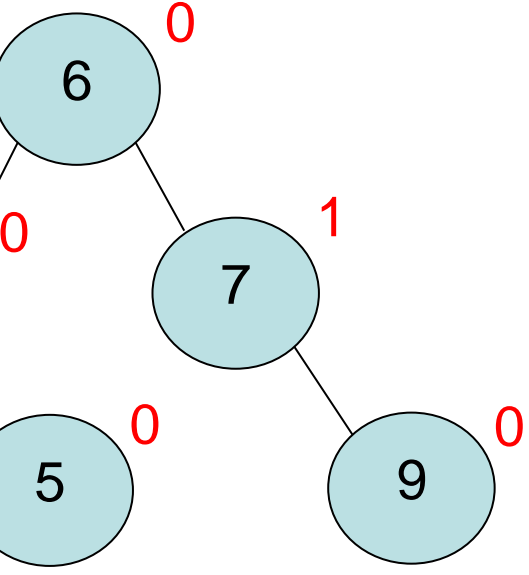


Aufgabe c.)

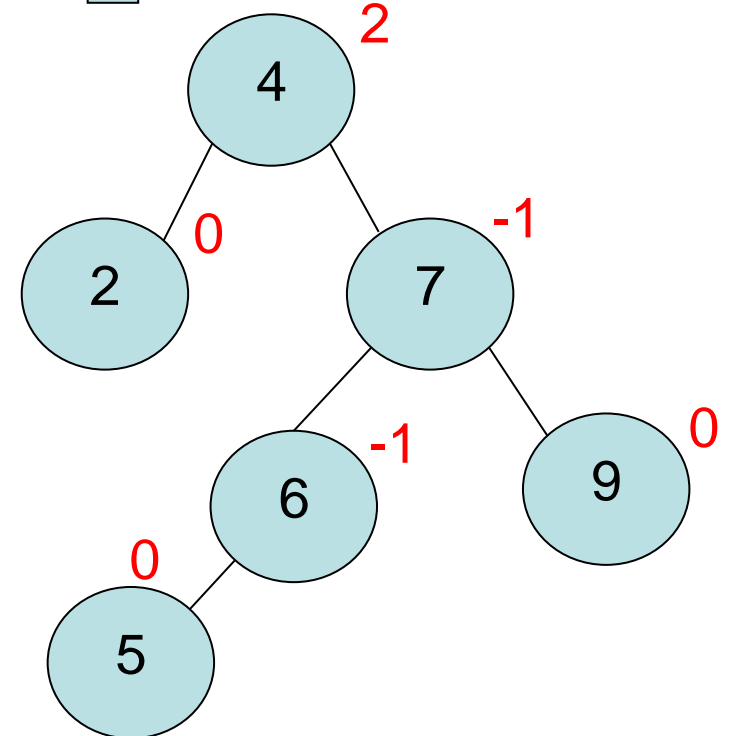
Unausgeglichener Baum:
Rechtsrotation



Einfügen
von:
9, 6, 5:



Unausgeglichener Baum:
Recht-Links-Rotation



Aufgabe d.)

- Kleinstes Element befindet sich immer am weitesten links
- Achtung: Spezialfall: Leerer Baum (root = null)

```
K minKey() {  
    Node<K> n = root;  
  
    if (n == null ) {  
        return null; // Oder Exception, je nach API  
    }  
  
    while (n.left != null) n = n.left;  
  
    return n.key;  
}
```

Aufgabe e.)

- Rekursive Lösung

```
int size() {  
    return size(root);  
}  
  
int size(Node<K> n) {  
    if (n == null) {  
        return 0;  
    }  
  
    return size(n.left) + size(n.right) + 1;  
}
```