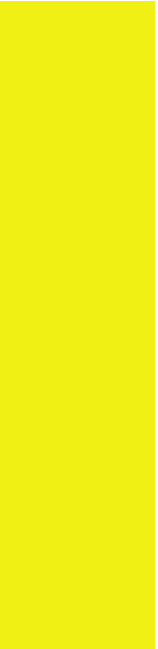


04 Bäume - Besprechung Kurztest

Algorithmen und Datenstrukturen 2



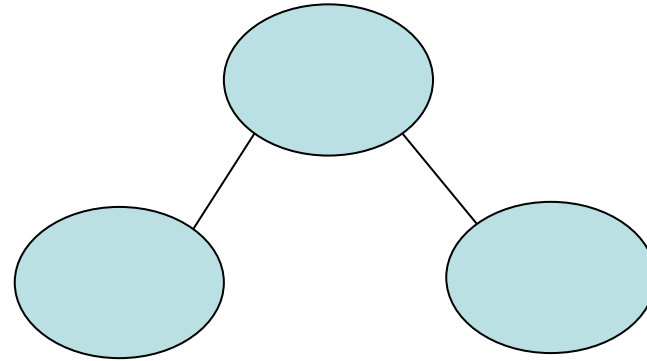
Aufgabe a.)

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

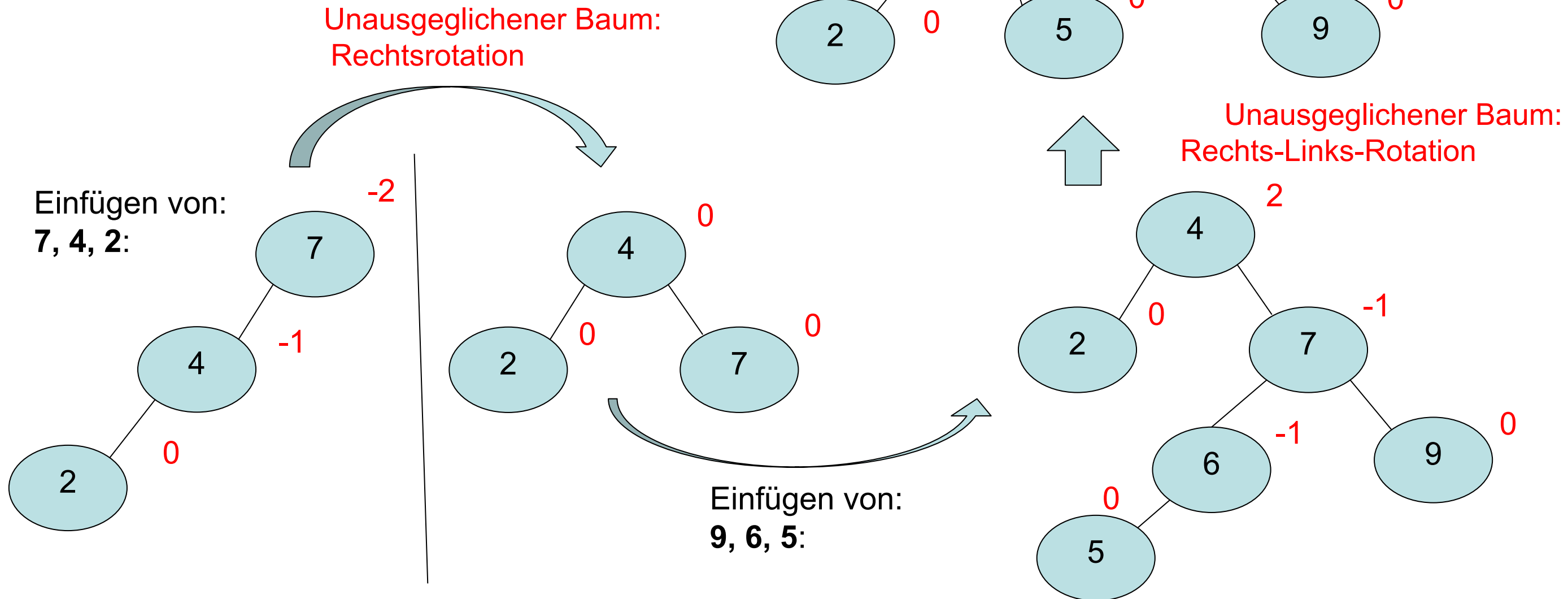
| Operation | Linked List | Binärer Suchbaum | AVL Baum |
|------------------|-------------|------------------|-------------|
| add(element) | $O(n)$ | $O(n)$ | $O(\log n)$ |
| contains(object) | $O(n)$ | $O(n)$ | $O(\log n)$ |
| remove(element) | $O(n)$ | $O(n)$ | $O(\log n)$ |

Aufgabe b.)

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



Aufgabe c.)



Aufgabe d.)

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

```
public 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

```
public int size() {  
    return size(root);  
}
```

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