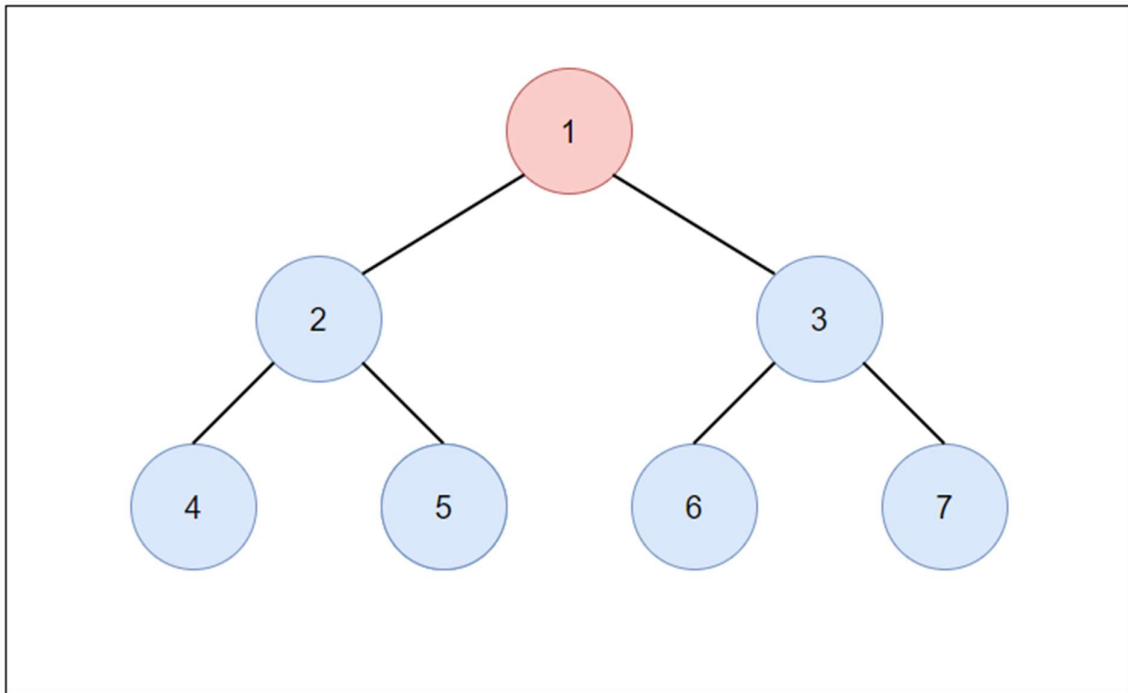


[A06] Tiefensuche

Grafische Beschreibung



Textuelle Beschreibung

Bei der Tiefensuche handelt es sich um ein fast ähnliches Verfahren wie bei der Breitensuche. Es wird hierbei zuerst ein Pfad vollständig in die Tiefe abgearbeitet. Dabei werden alle Knoten im Graphen abgearbeitet.

Der Ablauf der Tiefensuche ist folgender:

- Starte beim Root Knoten
- Suche nach Kindknoten von oben nach unten Ebene für Ebene durch, danach auf der anderen Seite
- Speichere die Kindknoten in einer „Queue“ zwischen und füge diese dann in eine Liste „besuchte Knoten“
- Wiederhole dies solange bis alle Knoten besucht wurden und alle zwischengespeicherten Knoten von der „Queue“ in die Liste übertragen wurden

Laufzeit

Durchschnittsfall: $O(|V| + |E|)$

Wobei $O(|E|)$ zwischen $O(V^2)$ und $O(1)$ wandert

Wichtige Elemente

```
public List<String> getNodesInOrder(Node<Film> node) {

    List<String> list = new ArrayList<>();    // create new list to store film title

    if (node == null)        // termination of recursion
        return list;

    list.addAll(getNodesInOrder(node.getLeft())); // call recursive method on left
    child leave
    list.add(node.getValue().getTitel());        // in the middle due the symmetry
    for in-order
    list.addAll(getNodesInOrder(node.getRight())); // call recursive method on right
    child leave

    return list;
}

/**
 * Retourneiert Titelliste jener Filme, deren Länge zwischen min und max liegt, in
 * Hauptreihenfolge (engl. pre-order, d.h. Knoten-links-rechts)
 * @param min Minimale Länge des Spielfilms
 * @param max Maximale Länge des Spielfilms
 * @return Liste der Filmtitel in Hauptreihenfolge
 */
// Big O notation -> O() */ //TODO - ADD O notation
public List<String> getMinMaxPreOrder(double min, double max) {
    //printTree();
    return getMinMaxPreOrder(min, max, root); // call method recursive with min &
    max from method parameter and the root
}

public List<String> getMinMaxPreOrder(double min, double max, Node<Film> node) {
    // overload method to call it recursive

    List<String> list = new ArrayList<>();    // create new list to store film title

    if (node == null)        // termination of recursion
        return list;

    if (node.getValue().getLänge() > min && node.getValue().getLänge() < max) //
    check if duration is within min & max parameter
        list.add(node.getValue().getTitel());        // if it is
    add to list

    list.addAll(getMinMaxPreOrder(min, max, node.getLeft()));        // call
    method recursive on left leave
    list.addAll(getMinMaxPreOrder(min, max, node.getRight()));        // call
    method recursive on right leave

    return list;
}
```