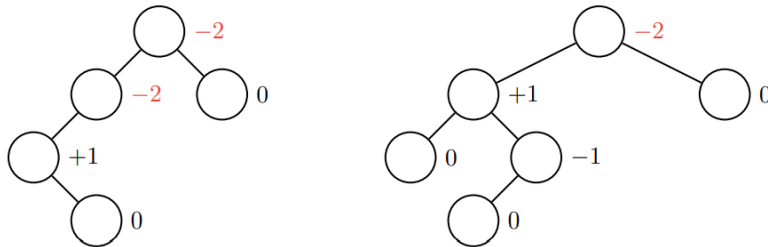


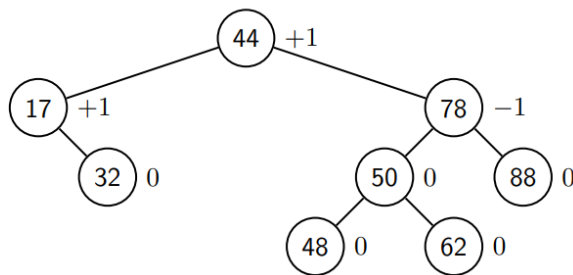
## 04 Bäume – Arbeitsblatt 2

1. Zeichnen Sie bei den folgenden Binärbäumen die Balancefaktoren aller Knoten ein und bestimmen Sie, ob diese ausgeglichen sind.



Beide Bäume sind nicht AVL-ausgeglichen.

2. Bestimmen Sie die Balancefaktoren aller Knoten des folgenden binären Suchbaumes. Handelt es sich um einen AVL-Baum?



Dies ist ein korrekter AVL-Baum, weil alle Knoten ausgeglichen sind.

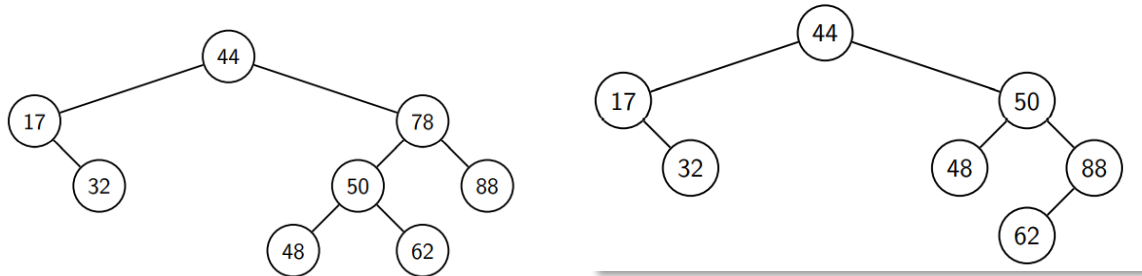
3. Schreiben Sie eine Funktion `boolean isbalanced(v)`, die `true` zurück gibt, falls der Binärbaum mit Wurzel `v` balanciert ist. Gerne dürfen Sie diese Funktion auch in ihre Java-Projekt integrieren und graphisch testen.

```
public boolean isbalanced() {return isbalanced(root) >= 0;}

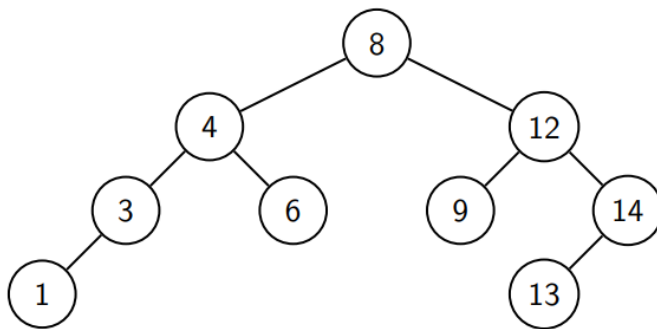
private int isbalanced(Node<K, E> t) {
    if (t != null){
        int h1 = height(t.left), hr = height(t.right);
        if (h1 == -1 || hr == -1) return -1;
        int diff = h1-hr;
        if (Math.abs(diff) > 1) return -1;
        return h1 >= hr ? h1 + 1 : hr + 1;
    } else return 0;
}
```

4. Entfernen Sie den Knoten mit Schlüssel 78 aus dem folgenden Suchbaum und führen Sie danach die notwendigen Rebalancierungen durch, um wieder einen AVL-Baum zu erhalten.

Rechts der resultierende AVL-Baum:



5. Zeichnen Sie den AVL-Baum, der durch Einfügen der Schlüssel 9, 14, 12, 4, 8, 6, 3, 1, 13 der Reihe nach in den anfangs leeren Baum entsteht. Führen Sie dabei die benötigten Rebalancierungen durch.



Alle Zeichnungen von: <https://educ.ethz.ch/unterrichtsmaterialien/informatik/binaere-suchbaeume.html>