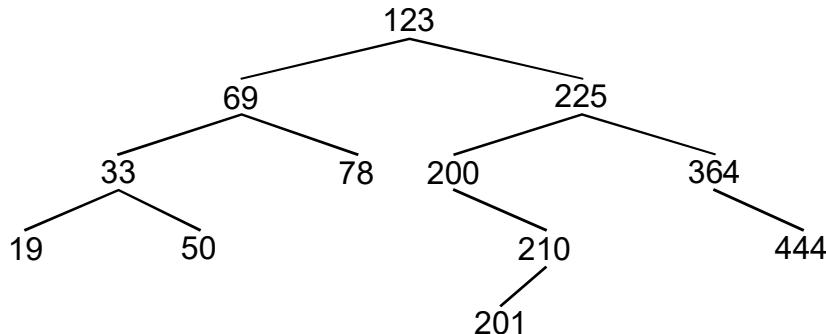


**VL06, Aufgabe 1 (Übung)**

Gegeben ist folgender Baum:



- Welche Höhe hat dieser Baum?
- Wie viele Blätter besitzt dieser Baum?
- Handelt es sich um einen Binärbaum?
- Handelt es sich um einen Suchbaum?
- Suchen Sie im dargestellten Baum nach den Zahlen 210, 50, 123 und 371.
- Ist der dargestellte Baum vollständig? Falls nicht, stellen Sie den Baum derart um bzw. fügen weitere Blätter derart hinzu, dass ein vollständiger Baum entsteht.

**VL06, Aufgabe 2 (Übung)**

Wenden Sie für die folgenden Aufgaben die Methoden `einfuegen` und `entfernen` der Klasse `Baum.java` aus `BSP06-Suchbaum.zip` auf einen anfangs leeren Suchbaum an:

- Fügen Sie nacheinander Knoten mit den Schlüsseln E, D, S, O, Z, P, A, R und Q ein.
- Traversieren Sie den so erzeugten Baum in Pre-Order, In-Order und Post-Order, und geben Sie an, in welcher Reihenfolge die Schlüssel ausgegeben werden.
- Löschen Sie dann nacheinander die Knoten mit den Schlüsseln D, S und E.

Zeichnen Sie den Baum, der durch das Einfügen aller Schlüssel entsteht, sowie die Bäume, die nach jedem einzelnen Löschvorgang entstehen.

**Hinweis:** Beachten Sie, dass das Programm zu löschen Knoten mit zwei Nachfolgern durch den größten Knoten im linken Teilbaum ersetzt.

## VL06, Aufgabe 3 (Praktikum)

Implementieren Sie in der Klasse `Baum` aus `BSP06-Suchbaum.zip` die folgenden drei Methoden:

- a) `private String traversierePreOrder(final Knoten<T> einKnoten)`
- b) `private String traversiereInOrder(final Knoten<T> einKnoten)`
- c) `private String traversierePostOrder(final Knoten<T> einKnoten)`

zum Durchlaufen des Baums in Pre-Order, In-Order und Post-Order. Setzen Sie dazu in jeder der drei Methoden einen `String` auf geeignete Weise zusammen (ggf. mittels einer lokalen Variable), den Sie an den Aufrufer zurückgeben.

**Hinweis:** die anderen Methoden dürfen nicht verändert werden! Ebenso dürfen Sie keine Attribute zur Klasse `Baum` hinzufügen!

Wie müssen Sie die Methode `traversiereInOrder` verändern, um die Knoten in umgekehrter Reihenfolge auszugeben?

## VL06, Aufgabe 4 (Praktikum)

Implementieren Sie in der Klasse `Baum` aus `BSP06-Suchbaum.zip` die Methode `private int hoeheRek(final Knoten<T> einKnoten)`, welche rekursiv die die Höhe des Baumes mit der Wurzel `einKnoten` bestimmen soll. Überlegen Sie vorab, wie die Rekursionsbasis und die Rekursionsvorschrift für die Berechnung der Höhe eines Binärbaumes aussieht.

**Hinweis:** Die anderen Methoden dürfen nicht verändert werden! Ebenso dürfen Sie keine Attribute zur Klasse `Baum` hinzufügen!