Binäre Suchbäume

Suchen

```
algorithm search(v, k)
    {Im Baum mit Wurzel v wird der Schlüssel k gesucht}
    if v != null then
        if k < key(v) then
            search(left(v), k) {Suche im linken Teilbaum left(v)}
        else
                if   k > key(v) then
                    search(right(v), k) {Suche im rechten Teilbaum right(v)}
                else
                    Beende Suche {Suche war erfolgreich}
                end if
        end if
    else
        Beende Suche {Suche war erfolglos}
    end if
end algorithm
```

Dieser Algorithmus wird nur den *ersten* Knoten mit dem Schlüssel *k* suchen und auch finden, falls er vorhanden ist. Hat es jedoch weitere Knoten mit diesem Schlüssel, werden diese nicht gefunden.

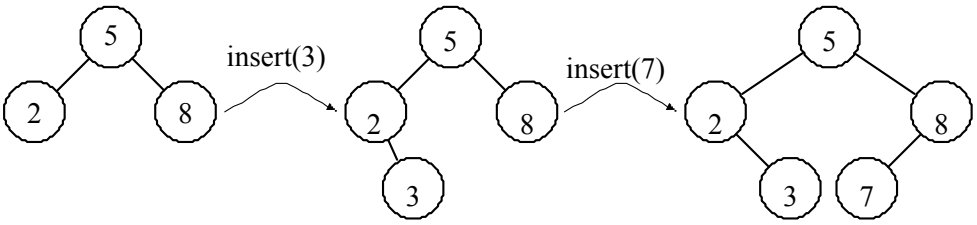

Abbildung: Das Einfügen eines Knotens mit Schlüssel 3 und 7 in den binären Suchbaum.

Einfügen eines neuen Knotens

Das **Einfügen** (insert) eines Knotens erfordert das Suchen der richtigen Einfügestelle im Baum. Dabei wird ähnlich vorgegangen wie beim Algorithmus search(). In jedem Knoten wird eine Vergleichsoperation ausgeführt und entschieden, ob der einzufügende Knoten im linken oder rechten Teilbaum gespeichert werden soll. Dieser Vergleich wird rekursiv ausgeführt bis der jeweilige Teilbaum leer ist. Hier ist nun der freie Platz, an dem der Knoten mit den Daten eingefügt werden kann.

Der folgende Algorithmus insert(*root, key*) in Pseudocode fügt einen Knoten mit dem Schlüssel *k* in den binären Suchbaum ein.

```
algorithm insert(v, k)
    { Im Baum mit Wurzel v wird nach einem Platz gesucht, an dem der
      neue Knoten mit Schlüssel k gespeichert werden kann.}

    if v != null then
        if k < key(v) then
            insert(left(v), k) {Knoten muss im linken Teilbaum gespeichert werden}
        else
            insert(right(v), k) {Knoten muss im rechten Teilbaum gespeichert werden}

        end if
    else
        {Füge hier den neuen Knoten mit Schlüssel k ein.}
    end if
end algorithm
```

Die Reihenfolge in der Knoten mit ihren Daten und Schlüsseln in einen binären Suchbaum einfügt werden, bestimmt die Struktur des Baumes. Leer bedeutet, dass der Teilbaum keine Knoten hat, also **null** ist.