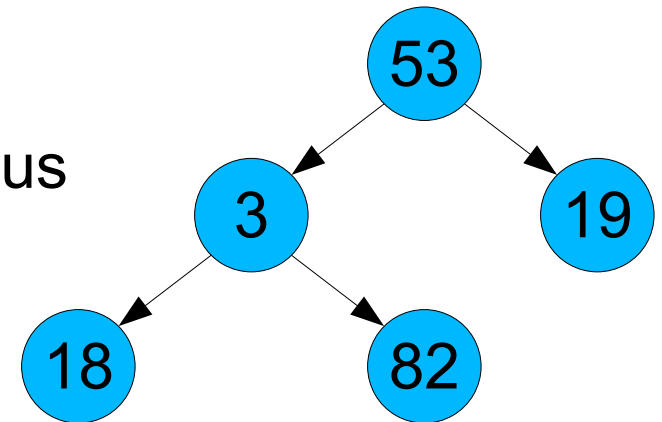


Algorithmen auf Bäumen

Eigenschaften von Bäumen lassen sich oft rekursiv definieren.

- **Anzahl** der Elemente (im Beispiel 5):
 - Ein leerer Baum hat 0 Elemente.
 - Jeder andere Baum hat 1 Element plus die Anzahl der Elemente in seinen beiden Kindern.
- **Tiefe** (im Beispiel 3): Anzahl der Knoten im längsten Pfad von der Wurzel zu einem Blatt
 - Ein leerer Baum hat die Tiefe 0.
 - Jeder andere Baum hat die Tiefe 1 plus die Tiefe seines „tieferen“ Kindes.



- **Suche** nach einem Wert x :
 - Ein leerer Baum enthält x nicht.
 - Andernfalls: Wenn der Datenwert des Baums gleich x ist, enthält der Baum x .
 - Andernfalls: Wenn mindestens einer der beiden Kinder x enthält, enthält der Baum x .
- **Summe der Werte:** (vorausgesetzt, die enthaltenen Daten sind Zahlen)
 - Die Summe eines leeren Baums ist 0.
 - Andernfalls ist die Summe der Datenwert plus die Summe der Kinder.

Algorithmen auf Bäumen

Die Implementation ist am einfachsten rekursiv umzusetzen.

Summe aller Zahlen (Pseudocode)

```
summe (t) :  
    wenn t == null:  
        return 0  
    sonst:  
        return summe (t.links) +  
                t.daten +  
                summe (t.rechts)
```