

Informatik I

Uebung 9

Gruppe A, ETZ E91

Patrick Boenzli

dev.orxonox.net/wiki/boenzlip/inf1

Lernziele

Die Studenten:

- wissen was ein **binary tree** ist
- koennen einen binary tree **implementieren** und **durchsuchen**

Binary Search Tree

a) Begriffserklärung:

- **Tree data structure**, jeder node (Blatt) hat max **2 Nachfolger** (binary)
- Nachfolger werden ueblicherweise mit **left/right** bezeichnet
- Elemente in einem tree koennen **geordnet** sein: z.B. links=kleiner
- Aehnlichkeiten zu **linked lists: !rekursives Programmieren!**

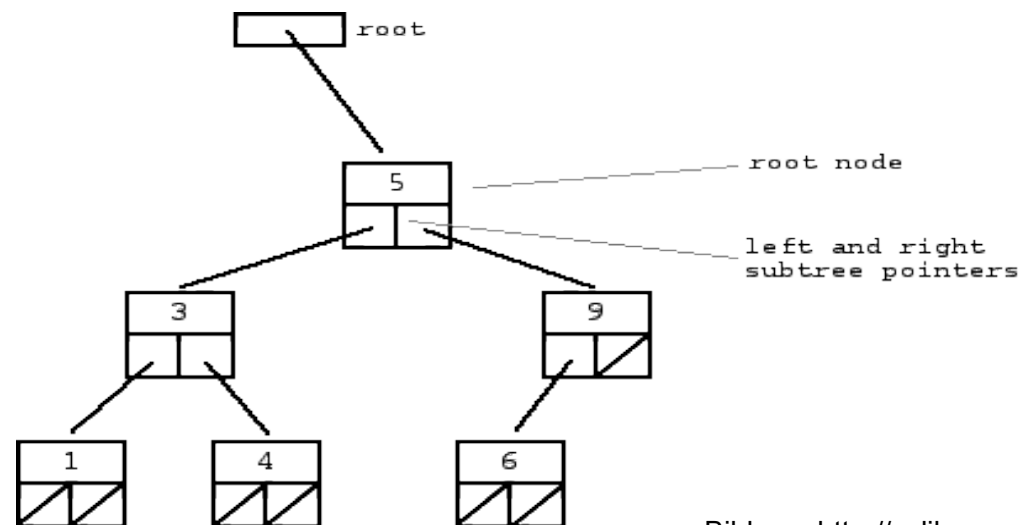


Bild aus <http://cslibrary.stanford.edu/110/BinaryTrees.html>

Binary Search Tree

b) Datenstruktur

```
// tree element: node
//
struct node
{
    int    value;    // the value of the node

    node*  left;    // reference to left neighbor
    node*  right;   // reference to right neighbor
};
```

Binary Search Tree

c) Rekursive Algorithmen

- Suchalgorithmen in trees sind meistens rekursiv

```
// example for a simple recursive search function
//
bool searchValue( int value, node* currentNode)
{
    if( currentNode == NULL) return false;
    if( someCriterion == true) return true;

    if( searchValue(value, currentNode->left))
        return true;
    if( searchValue(value, currentNode->right))
        return true;
}
```

Aufgabe 1

a) Aufgabe verstehen

- 1) Datenstrukturen fuer Binary Search Tree schreiben
- 2) Funktion die rekursiv durch den Baum geht
- 3) Einfuegen/Loeschen von Elementen im Baum
- 4) Suchfunktion durch den Baum

Aufgabe 2

a) Aufgabe verstehen

- 1) Einlesen der binären Daten wie in Aufg. 7 (siehe MuLoe7)
- 2) Hinzufügen der Daten in den Baum durch die Funktion in Aufg 1c)
- 3) Suchfunktion wie in Aufg 1b)