



Übersichtsblatt: Binäre Suchbäume

Definition:

Ein Binärer Suchbaum ist eine dynamische Datenstruktur die nach dem folgenden Prinzip aufgebaut ist. Ein Baum bei dem jeder Knoten ≤ 2 Kinder besitzt (ein so genannter Binärbaum) wird derart modifiziert, dass in den Knoten des Baumes nicht nur die Daten, sondern auch noch ein Schlüsselwert gespeichert werden. Dies ermöglicht das effiziente Suchen diesem Baum. Eine solche Datenstruktur nennt man auch Wörterbuch oder Dictionary.

Implementierung in C:

```
struct Knoten{
    <typ> key;           /*Schlüssel*/
    <typ> data;         /*Daten*/
    struct Knoten *left; /*linker Teilbaum*/
    struct Knoten *right; /*rechter Teilbaum*/
}
```

Organisationsprinzip:

- Der Knoten k enthält den Schlüssel k.key
- Alle Schlüsselwerte im linken Teilbaum k.left sind kleiner als k.key.
- Alle Schlüsselwerte im rechten Teilbaum k.right sind größer als k.key.

Operationen:

Gemäß der obigen Definition eines Binären Suchbaumes sind die folgenden Operationen auf dieser Datenstruktur ausführbar.

Im Suchbaum Suchen:

1. Wenn der Baum leer ist, wird NULL zurück gegeben.
2. Ist der Schlüssel eines Knotens kleiner als der gesuchte Schlüssel, wird im rechten Teilbaum weitergesucht.
3. Ist der Schlüssel eines Knotens größer als der übergebene Schlüssel, wird im linken Teilbaum weitergesucht
4. Dies wird wiederholt bis der Schlüssel gefunden wurde, oder man an einem Blatt ankommt, welches nicht den gesuchten Schlüssel enthält, dann wird NULL zurückgegeben.

Zeitaufwand:

Ausgeglichener Baum:	$O(\log(n))$
Entarteter Baum:	$O(n)$

Einfügen in einen Suchbaum:

1. Wenn der Baum leer ist, wird der einzufügende Knoten zur Wurzel des Baumes.
2. Wenn der Baum bereits Knoten enthält, so muss gemäß der o. g. Organisationsstruktur der Knoten gefunden werden, der der Elternknoten des einzufügenden Knoten wird.

Zeitaufwand:

Ausgeglichener Baum:	$O(\log(n))$
Entarteter Baum:	$O(n)$

Entfernen aus dem Suchbaum:

1. Wenn der zu entfernen Knoten k ein Blatt ist, muss nur der entsprechende Zeiger des Elternknotens auf Null gesetzt werden, dann kann der Knoten gelöscht werden.
2. Hat der zu entferne Knoten k ein Kind, so muss der Zeiger des Elternknotens auf den entsprechenden left oder right Zeiger von k gesetzt werden bevor k gelöscht werden kann.
3. Hat der Knoten zwei Kinder, so muss aus dem linken Teilbaum das größte Element herausgesucht werden. Der Elternzeiger von k wird auf dieses größte Element umgehängt. Die left und right Zeiger des großen Elementes werden auf den linken und rechten Teilbaum gesetzt. Jetzt kann k gelöscht werden.

Zeitaufwand:

Ausgeglichener Baum:	$O(\log(n))$
Entarteter Baum:	$O(n)$

