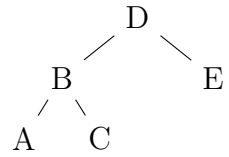
Traversierung

Es gibt Fälle, in denen man alle Elemente eines Baumes in eine lineare Liste bringen möchte. Dabei kommt es entscheidend darauf an, in welcher Reihenfolge die Elemente im Baum in die Liste eingefügt werden. Die unterschiedlichen Reihenfolgen führen zu verschiedenen Arten der sogenannten Traversierung. Zur Verdeutlichung wird der folgende Baum herangezogen:



Pre-Order (W-L-R) Bei dieser Traversierung, auch Tiefensuche genannt, wird zuerst die Wurzel (W) betrachtet. Anschließend wird der linke (L) und dann der rechte (R) Teilbaum durchlaufen. Das Beispiel liefert dann: D B A C E.

Post-Order (L–R–W) Diese trägt auch den Namen Nebenreihenfolge. Bei ihr wird als erstes der linke (L) und dann der rechte (R) Teilbaum durchlaufen. Zum Schluss wird die Wurzel (W) betrachtet. Das Beispiel liefert dann: A C B E D.

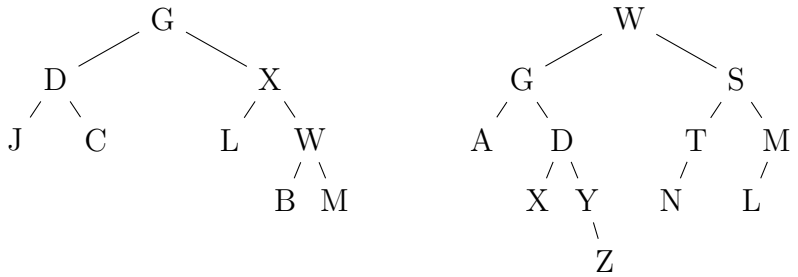
In-Order (L-W-R) Bei ihr spricht man auch von der symmetrischen Reihenfolge. Es wird der linke (L) Teilbaum durchlaufen. Danach folgt die Wurzel (W) und am Ende der rechte (R) Teilbaum. Das Beispiel liefert dann: A B C D E.

Reverse In-Order (R–W–L) Bei dieser anti-symmetrischen Reihenfolge wird zuerst der rechte (R) Teilbaum durchlaufen. Darauf folgt die Betrachtung der Wurzel (W) und am Ende der linke (L) Teilbaum. Das Beispiel liefert: E D C B A.

Level-Order Sie ist auch bekannt unter dem Namen Breitensuche. Bei ihr werden beginnend an der Baumwurzel alle Ebenen von links nach rechts durchlaufen. Das Beispiel liefert: D B E A C.

Aufgabe 1

Traversieren Sie die folgenden Bäume mit allen Traversierungsarten.



Algorithmus

Die ersten vier Arten der Traversierung lassen sich durch ähnliche Algorithmen realisieren. Bei ihnen muss nur die Reihenfolge der Elemente geändert werden. Hier ein Beispiel für In-Order:

traversiereInOrder(baum, liste):

baum.gibLinkerTeilbaum() ist nicht leer	
ja	nein
traversiereInOrder(baum.gibLinkerTeilbaum(), liste)	∅
liste.haengeAn(baum.gibWurzel())	
baum.gibRechterTeilbaum() ist nicht leer	
ja	nein
traversiereInOrder(baum.gibRechterTeilbaum(), liste)	∅

Aufgabe 2

Geben Sie die Reihenfolge der Codeblöcke des oberen Struktogramms an, um die anderen drei Traversierungsarten damit zu realisieren.

Aufgabe 3

Ein Baum wurde Post-Order traversiert. Das Ergebnis lautet G D V Z H K L Q W E R. Geben Sie einen Ursprungsbaum an, der dieses Ergebnis liefert.

Lösungen zu den Aufgaben

Lösung 1

Pre-Order G D J C X L W B M und W G A D X Y Z S T N M L

Post-Order J C D L B M W X G und A X Z Y D G N T L M S W

In-Order J D C G L X B W M und A G X D Y Z W N T S L M

Reverse In-Order M W B X L G C D J und M L S T N W Z Y D X G A

Level-Order G D X J C L W B M und W G S A D T M X Y N L Z

Lösung 2

In-Order (Gegeben) A,B,C

Pre-Order B,A,C

Post-Order A,C,B

Reverse In-Order C,B,A

Lösung 3

Für die Ausgabe gibt es mehrere Möglichkeiten, wie der Ausgangsbaum ausgesehen haben kann. Hier ist ein möglicher dargestellt:

