

Binärbäume

Traversierung

- Unter der **Traversierung** eines Baumes versteht man das systematische Durchlaufen und Untersuchen aller Knoten in einer bestimmten Reihenfolge
 - Jeder Knoten soll dabei nur ein Mal untersucht werden
- Sowohl Breitensuche als auch Tiefensuche erlauben die Traversierung beliebiger Bäume
- Die Traversierung binärer Bäume mittels Tiefensuche kann weiter untergliedert werden in
 - Pre-Order (Hauptreihenfolge)
 - In-Order (symmetrische Reihenfolge)
 - Post-Order (Nebenreihenfolge)

Traversierungsarten (1)

- Für jeden von der Tiefensuche besuchten Knoten werden die folgenden Schritte durchgeführt:
 - Untersuchen des Knotens (z. B. Suchen nach einem bestimmten Wert oder Wert ausgeben)
 - Rekursiver Abstieg in den linken Teilbaum, falls vorhanden
 - Rekursiver Abstieg in den rechten Teilbaum, falls vorhanden
- Die Reihenfolge dieser drei Schritte kann beliebig gewählt werden
 - Wir betrachten im folgenden nur Fälle, bei denen der linke Teilbaum vor dem rechten Teilbaum durchlaufen wird

Traversierungsarten (2)

- Pre-Order
 - Untersuche den aktuellen Knoten
 - Steige in den linken Teilbaum ab
 - Steige in den rechten Teilbaum ab
- In-Order
 - Steige in den linken Teilbaum ab
 - Untersuche den aktuellen Knoten
 - Steige in den rechten Teilbaum ab
- Post-Order
 - Steige in den linken Teilbaum ab
 - Steige in den rechten Teilbaum ab
 - Untersuche den aktuellen Knoten

Aufgabe 1

- Erweitere Dein Programm zur Arbeit mit binären Suchbäumen um Methoden zur Baumtraversierung auf die drei genannten Arten
 - Der “Untersuchungsschritt” für jeden Knoten soll in der Ausgabe seines Wertes bestehen
- Teste die Ausgabereihenfolge bei folgendem binären Suchbaum

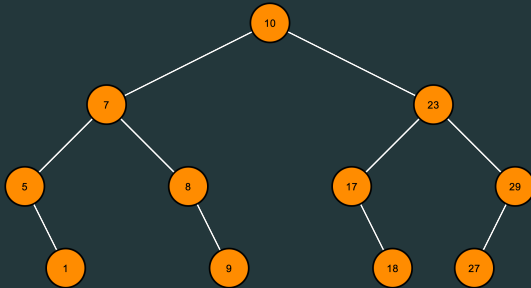


Abbildung 1: Binärer Suchbaum zum Testen

Speicherung als Array

- Binärbäume lassen sich als Array speichern
 - Für einen Binärbaum der Höhe h wird mindestens ein Array der Größe $2^{h+1} - 1$ benötigt
 - Die Eltern-Kind-Beziehung zwischen Knoten wird nicht explizit gespeichert
- Vorgehen:
 - Die Knoten werden als nummeriert angenommen: Die Wurzel mit 1, alle weiteren Knoten fortlaufend in der Reihenfolge der Breitensuche
 - Knoten i wird am Index i im Array gespeichert (der Platz an Index 0 bleibt unbenutzt)
 - Die Kinder von i befinden sich an den Positionen $2i$ und $2i + 1$ im Array
 - Der Elternknoten von i befindet sich an der Position $\lfloor \frac{i}{2} \rfloor$

Aufgabe 2

- Skizziere den Inhalt eines Arrays zur Speicherung des Beispiel-Binärbaums aus Aufgabe 1 gemäß dem vorgestellten Schema
 - Prüfe anhand der Skizze die Berechnungsvorschriften zum Finden von Kind- und Elternknoten