

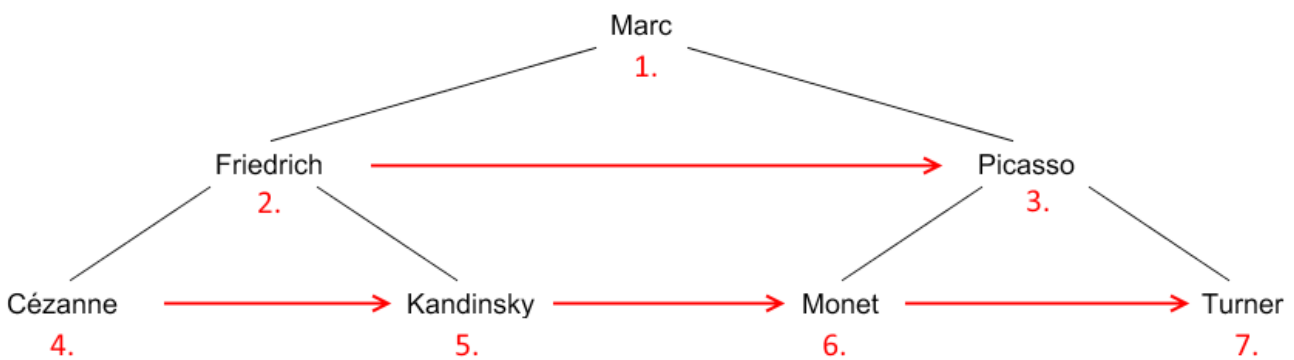
Verschiedene Reihenfolgen

„Traversieren“ bedeutet, alle Knoten eines (Binär-)baums einmal zu „besuchen“ und dabei etwas mit diesen Knoten zu tun: ausgeben, mit einem Suchbegriff vergleichen, zählen, auswerten usw.

Es gibt verschiedene Reihenfolgen, in denen ein Binärbaum traversiert werden kann. Je nach Anwendung sind nur bestimmte Reihenfolgen sinnvoll. Die Reihenfolgen können zunächst unterteilt werden in die sogenannte Breiten- und Tiefensuche (man spricht von „Suche“, weil das eine häufige Anwendung der Traversierung ist, aber diese Methoden werden auch für andere Anwendungen eingesetzt).

Breitensuche

Bei der Breitensuche beginnt man an der Wurzel, besucht dann alle Knoten der Tiefe 2, dann alle Knoten der Tiefe 3, usw. Man durchwandert den also Baum immer in der ganzen Breite, bevor man eine Tiefe weiter nach unten geht.



Diese Reihenfolge ist z.B. geeignet, um einen binären Suchbaum **in einer Datei zu speichern** – wenn man den Baum nach Beenden des Programms aufbewahren möchte, um ihn zu einem späteren Zeitpunkt wieder zu verwenden. Eine Datei hat im Prinzip die Struktur einer linearen Liste: es wird ein Objekt nach dem anderen in die Datei geschrieben. Die Struktur des Baums muss also in eine lineare Reihenfolge gebracht werden¹.

Wenn man später aus dieser linearen Struktur den Suchbaum von neuem aufbaut, hat er exakt die gleiche Struktur wie das Original.

Die Breitensuche ist einfach zu verstehen, jedoch schwierig bzw. aufwendig zu implementieren, denn die Knoten einer Tiefe sind ja nicht direkt miteinander verbunden. Du solltest dir den Begriff Breitensuche merken, brauchst sie aber im Unterricht nicht zu programmieren.

¹ Man spricht auch von „serialisieren“, die Inhalte des Baums sozusagen in einer Serie aufzuschreiben

Tiefensuche

Bei der Tiefensuche wandert man entlang eines „Ast“ des Baums bis ganz unten zu einem der Blätter. Dann geht man ein Stück nach oben zurück und auf der anderen Seite wieder nach unten, dann wieder hoch, und so weiter, bis der ganze Baum traversiert wurde.

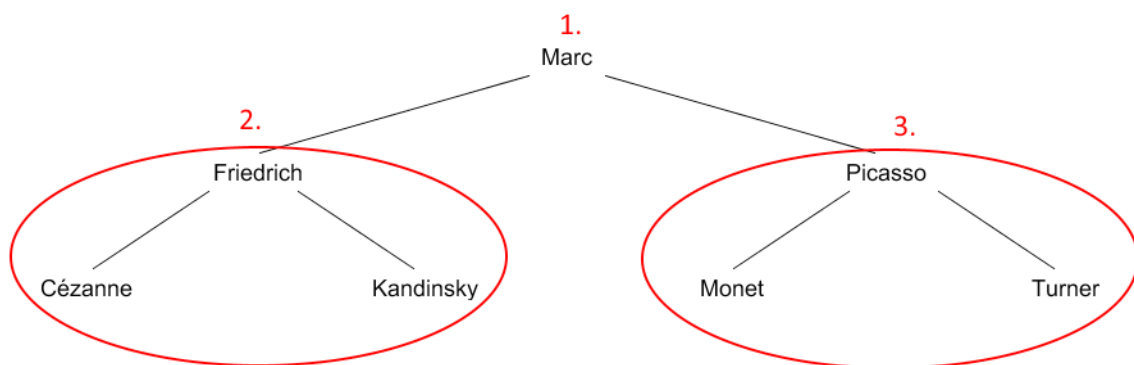
Es gibt mehrere Varianten der Tiefensuche, die sich in ihrer praktischen Anwendung deutlich unterscheiden. Das Unterscheidungsmerkmal ist, in welcher Reihenfolge jeweils ein Knoten und seine Teilbäume „besucht“ werden.

Pre-order-Reihenfolge

Die verschiedenen Reihenfolgen der Tiefensuche sind **rekursiv** zu verstehen.

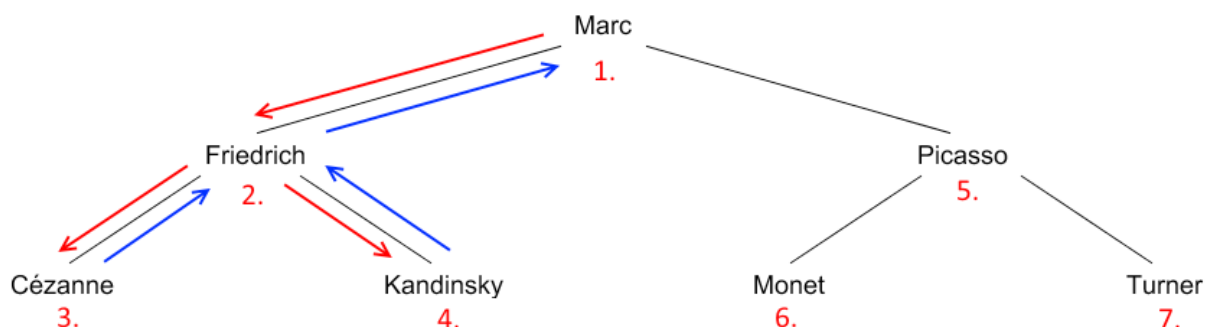
Bei der pre-order-Reihenfolge gilt für jeden Knoten:

Zuerst wird dieser Knoten bearbeitet, dann sein linker, und anschließend sein rechter Teilbaum.



Wie man in der Grafik sieht, wird erst die Wurzel („Marc“), dann der linke und dann der rechte Teilbaum bearbeitet. Für die Teilbäume gilt die gleiche Regel (deswegen „rekursiv“): Im linken Teilbaum wird zuerst „Friedrich“ bearbeitet, dann links „Cézannes“ und dann rechts „Kandinsky“.

Dadurch ergibt sich dann die folgende Reihenfolge:

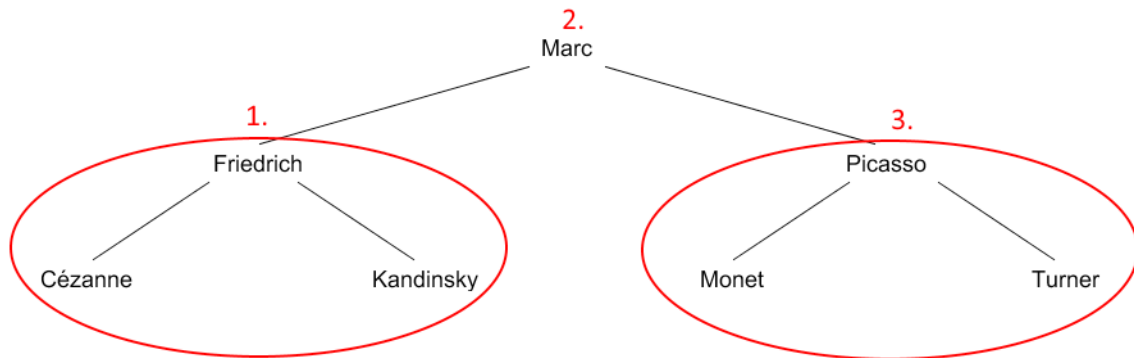


Man sieht, dass der Algorithmus nicht die Knoten einer Tiefe nacheinander bearbeitet, sondern immer ganz bis unten geht und dann wieder nach oben.

Die pre-order-Reihenfolge eignet sich wie die Breitensuche, um den Baum in einer linearen Liste (oder einer Datei) **abzuspeichern**, um ihn später genau wie vorher wieder aufzubauen.

In-order-Reihenfolge

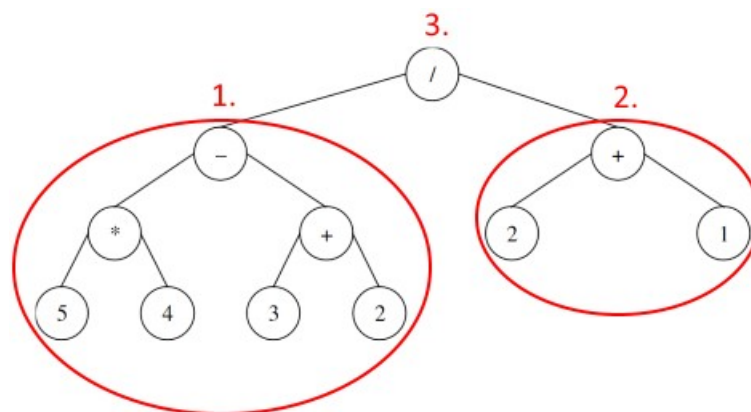
Bei der in-order-Reihenfolge gilt für jeden Knoten: Zuerst wird der linke Teilbaum des Knotens bearbeitet, dann der Knoten selbst, und danach sein rechter Teilbaum.



Wenn man einen binären Suchbaum mit der in-order-Reihenfolge traversiert, ergibt sich die **alphabetische Reihenfolge**.

Post-order-Reihenfolge

Bei der post-order-Reihenfolge gilt für jeden Knoten: Zuerst wird der linke Teilbaum, dann der rechte Teilbaum, und zum Schluss der Knoten selbst bearbeitet.



Ein typisches Beispiel für die post-order-Reihenfolge ist die **Auswertung eines Termbaums**.

Bevor man die Operation an der Wurzel (im Beispiel eine Division) ausführen kann, müssen erst die Ergebnisse der beiden Teilbäume vorliegen. Das gilt dann rekursiv auch wieder für die Teilbäume: im linken Teilbaum kann die Subtraktion erst ausgeführt werden, wenn die Ergebnisse von links (Multiplikation) und rechts (Addition) vorliegen.

Zusammenfassung

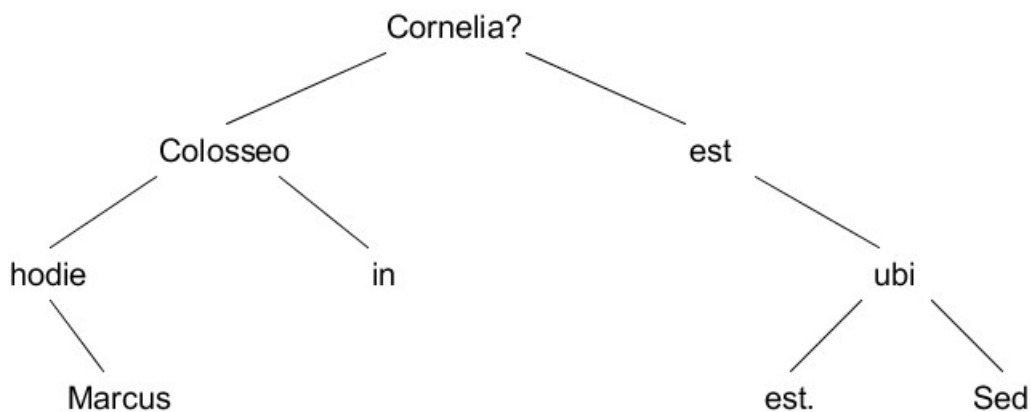
pre-order	→ Der Knoten zuerst :	Knoten , linker TB, rechter TB
in-order	→ Der Knoten in der Mitte :	linker TB, Knoten , rechter TB
post-order	→ Der Knoten zuletzt :	linker TB, rechter TB, Knoten

Aufgabe 1

Der folgende Binärbaum wird in den verschiedenen Reihenfolgen traversiert:

- Breitensuche
- pre-order (Tiefensuche)
- in-order
- post-order

Gib zu jeder Reihenfolge die Sequenz der Wörter an, die dabei entsteht.



Aufgabe 2

Statt Namen kann ein Suchbaum auch Zahlen verwalten. Es gilt die gleiche Regel: Für jeden Knoten gilt, dass alle Zahlen im linken Teilbaum kleiner sind als die Zahl am Knoten selbst, und alle Zahlen im rechten Teilbaum entsprechend größer.

Die pre-order-Traversierung eines Suchbaums lautet:

8, 4, 0, 2, 6, 14, 10, 12, 18, 16, 20

Zeichne den Suchbaum, aus dem diese Liste entstanden ist.