

Grundlagen der Informatik und Programmierung 1

Bäume

Jetzt auch noch Botanik?

Prof. Dr. Tom Vierjahn

Fachbereich Wirtschaft und Informationstechnik
Westfälische Hochschule – Campus Bocholt

Wintersemester 2019/20

Wiederholung: Einfach verkettete Liste

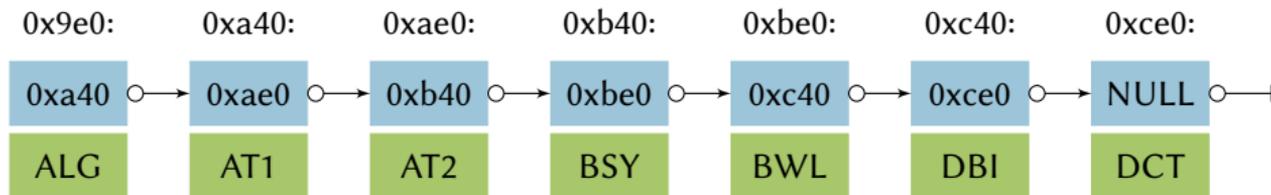


Liste

struct List modules	
first	0x9e0
last	0xce0

struct ListElement alg	
module	{., "ALG", .}
next_1	0x4a0
next_2	...

Speicher



struct Tree modules

root

&gip1

struct TreeNode gip1

module

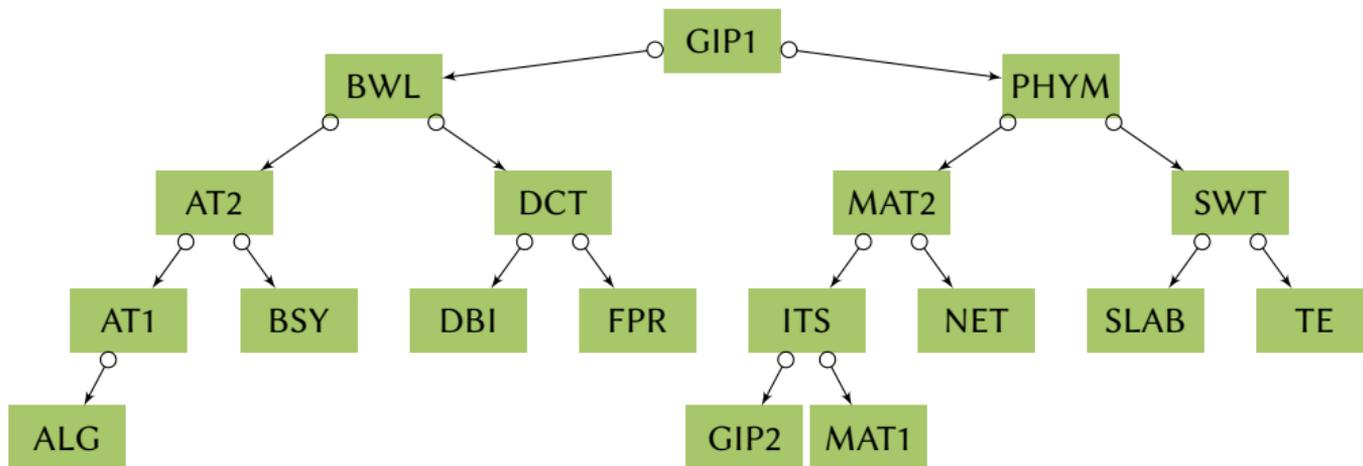
{., "GIP1", .}

left

&bw1

right

&phym



Definition: Baum

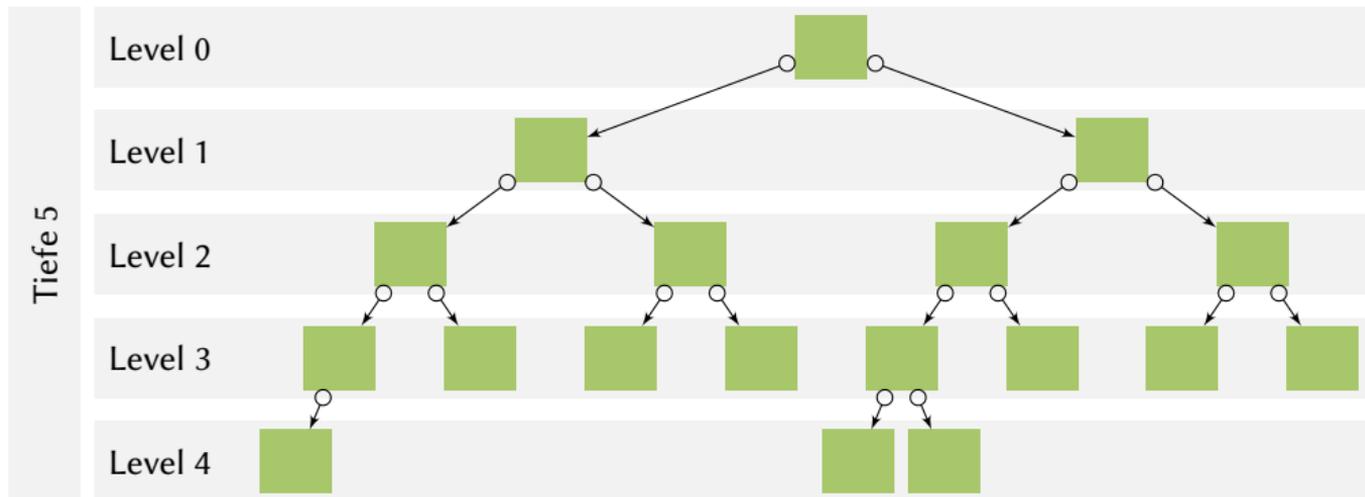
Ein **Baum** verallgemeinert den Listenbegriff: Jedes Element hat eine endliche Folge von **Nachfolgern**. Ein Baum hat keine Schleifen oder Zyklen. Ein Baum mit einer garantierten maximalen Höhe von $c \log n$ ist **balanciert**.

Definition: Knoten

Die Elemente eines Baumes heißen **Knoten**. In einem Baum gibt es genau einen Knoten ohne Vorgänger – den **Wurzelknoten** oder die **Wurzel**. Knoten, die keinen Nachfolger haben heißen **Blätter**.

Definition: Binärbaum

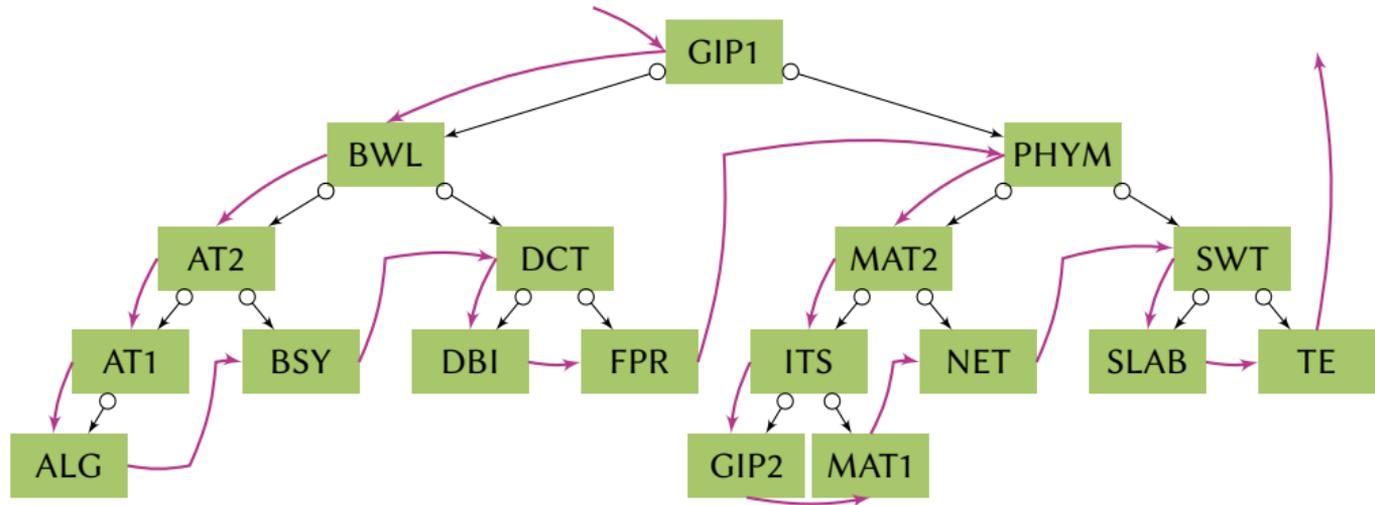
Die Knoten eines **Binärbaums** haben *maximal zwei* Nachfolger – den linken und den rechten.



- ▶ Level: Anzahl der Nachfolgeoperationen von der Wurzel bis zum Knoten
- ▶ Tiefe: maximales Level + 1
- ▶ pro Level i im Binärbaum: maximal 2^i Knoten
- ▶ Binärbaum der Tiefe t : maximal $2^t - 1$ Knoten
- ▶ Binärbaum mit n Knoten: Mindest-Tiefe $\log_2 n$

Pre-Order Traversierung

- ▶ erst der Knoten, dann links, dann rechts (NLR)

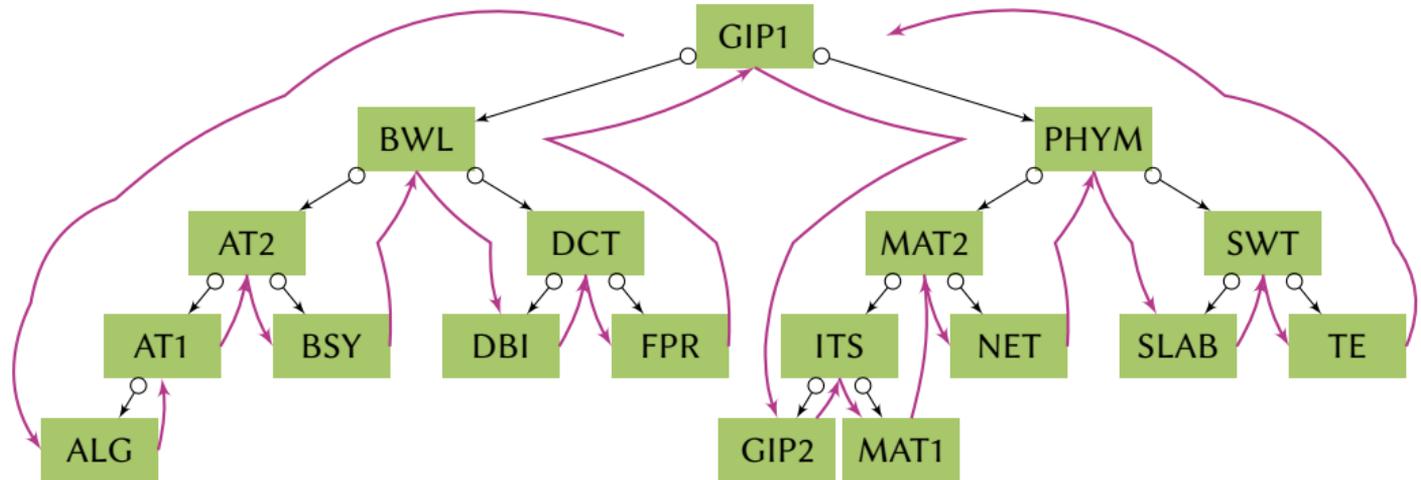


Ausgabe der Modulkürzel:

- ▶ GIP1, BWL, AT2, AT1, ALG, BSY, DCT, DBI, FPR, PHYM, MAT2, ITS, GIP2, MAT1, NET, SWT, SLAB, TE

In-Order Traversierung

- ▶ erst links, dann der Knoten, dann rechts (LNR)

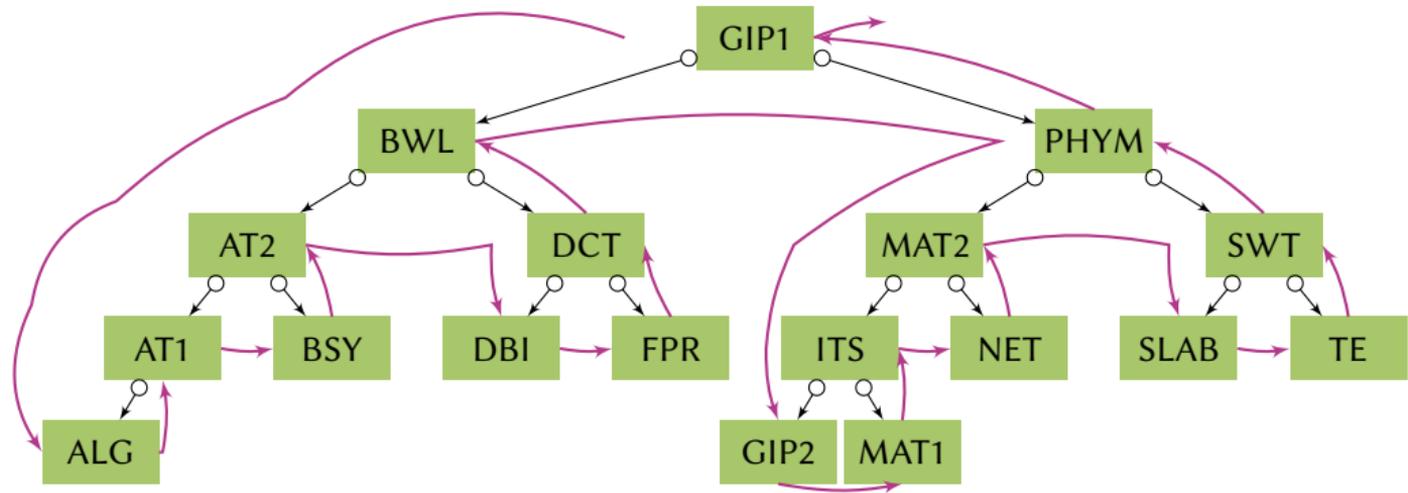


Ausgabe der Modulkürzel:

- ▶ ALG, AT1, AT2, BSY, BWL, DBI, DCT, FPR, GIP1, GIP2, ITS, MAT1, MAT2, NET, PHAM, SLAB, SWT, TE

Post-Order Traversierung

- erst links, dann rechts, dann der Knoten (LRN)

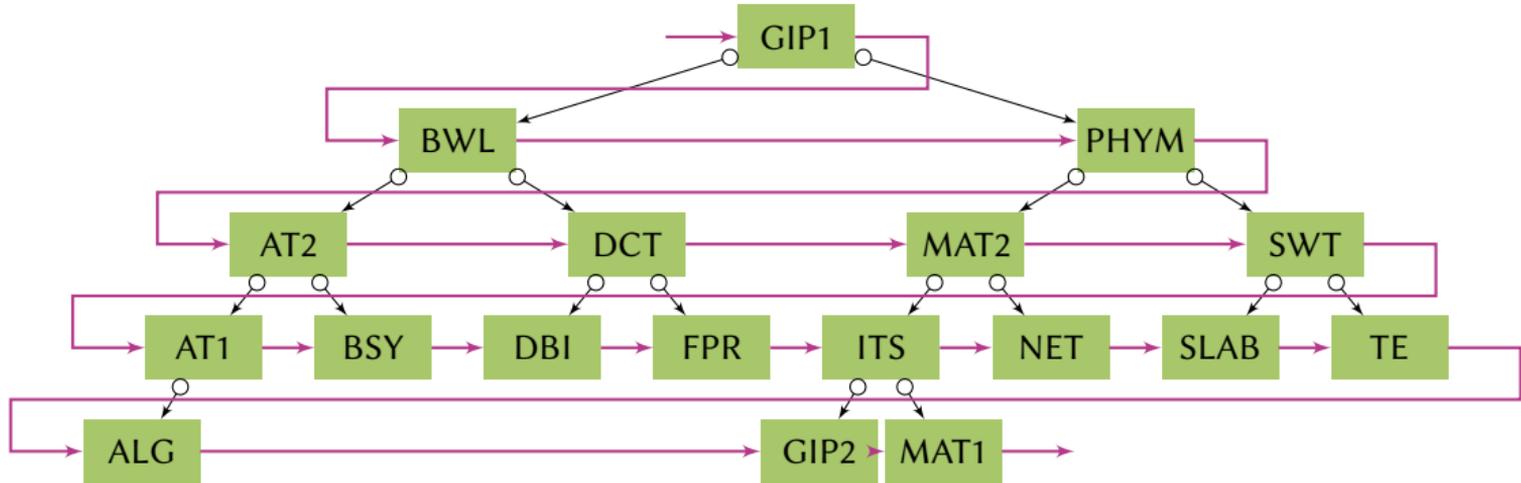


Ausgabe der Modulkürzel:

- ALG, AT1, BSY, AT2, DBI, FPR, DCT, BWL, GIP2, MAT1, ITS, NET, MAT2, SLAB, TE, SWT, PHYM, GIP1

Level-Order Traversierung

- ▶ erst alle Knoten einer Ebene



Ausgabe der Modulkürzel:

- ▶ GIP1, BWL, PHYM, AT2, DCT, MAT2, SWT, AT1, BSY, DBI, FPR, ITS, NET, SLAB, TE, ALG, GIP2, MAT1

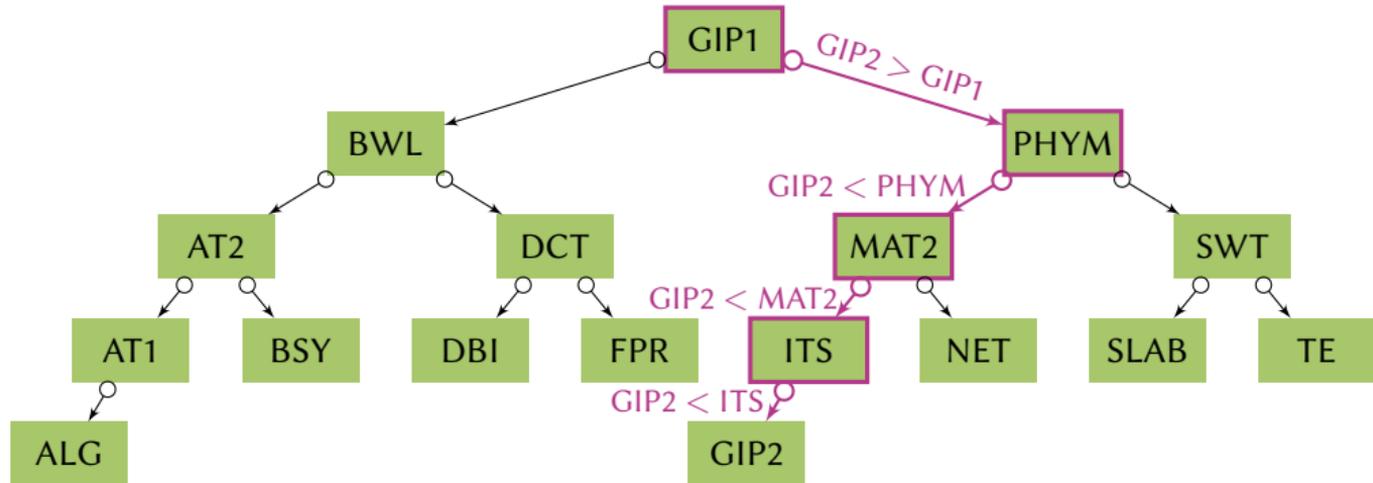
Tiefensuche (mit Rekursion/Stack)

- ▶ Pre-Order Traversierung: Kopieren des Baums
- ▶ In-Order Traversierung: Sortierte Verarbeitung der Knoten (bei sortierten Bäumen)
- ▶ Post-Order Traversierung: Abbau des Baumes

Breitensuche (mit Queue)

- ▶ Level-Order Traversierung: Kopieren des Baums, vorteilhaft bei Arrayrepräsentation

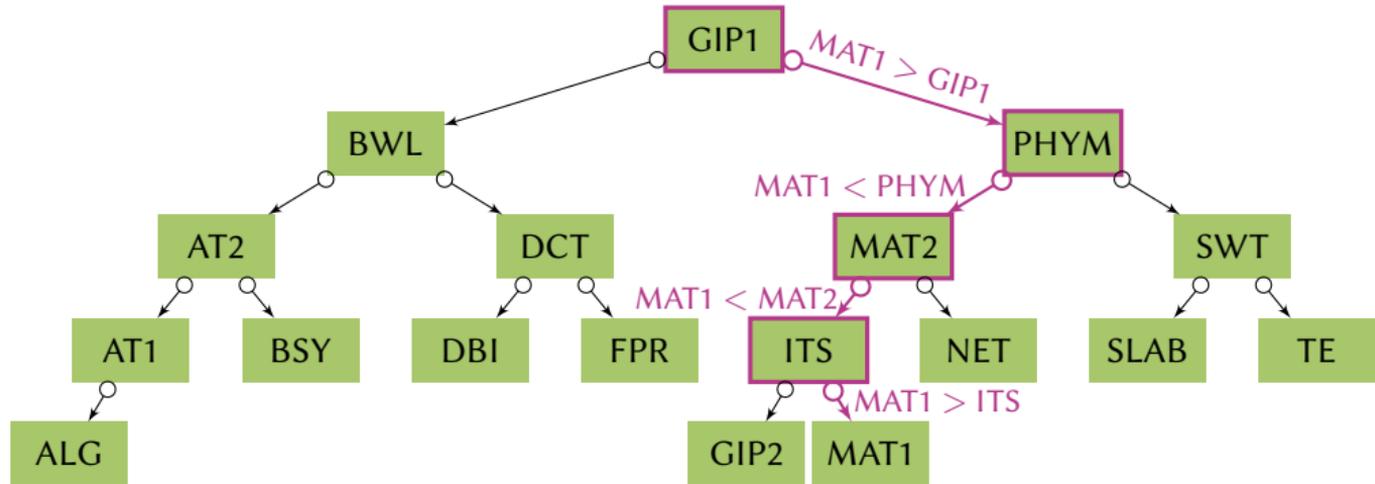
Sortiertes Einfügen



Einzufügendes Module:

- ▶ GIP2, MAT1

Sortiertes Einfügen



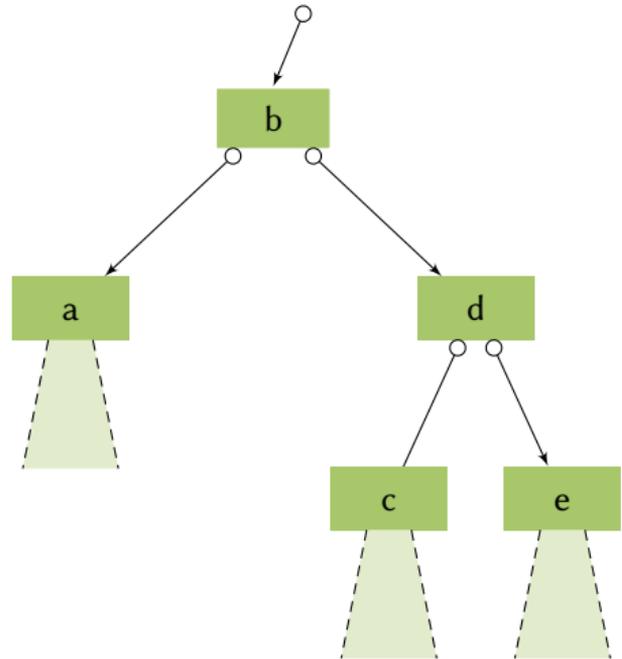
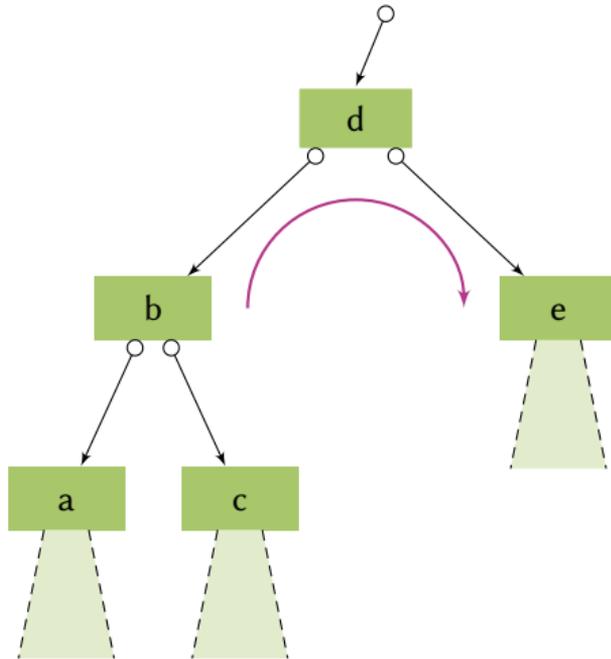
Einzufügendes Module:

- ▶ GIP2, MAT1

Rotationen

Knoten bewegen, ohne die Sortierung zu ändern

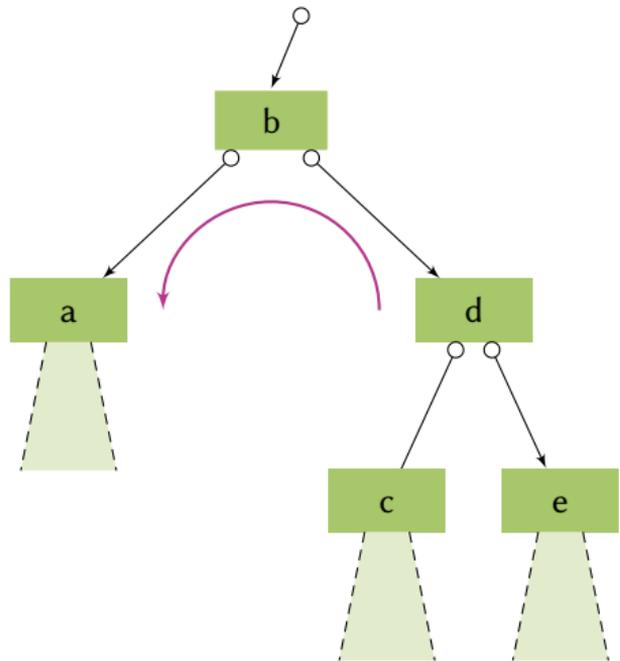
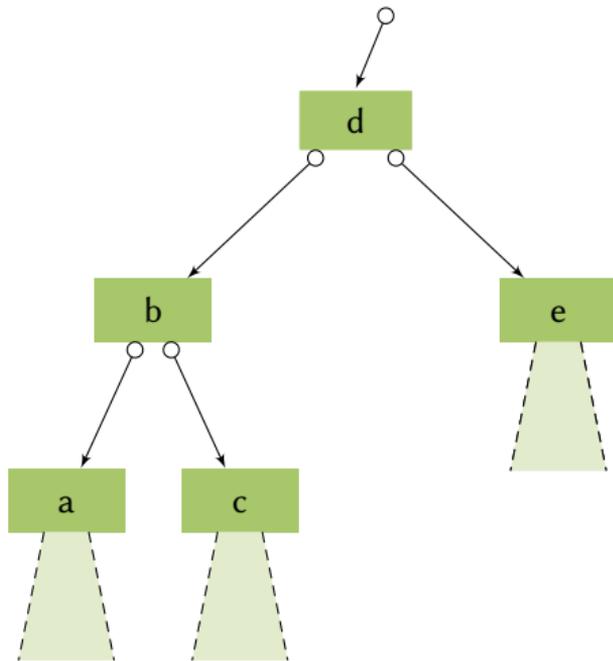
Rechtsrotation



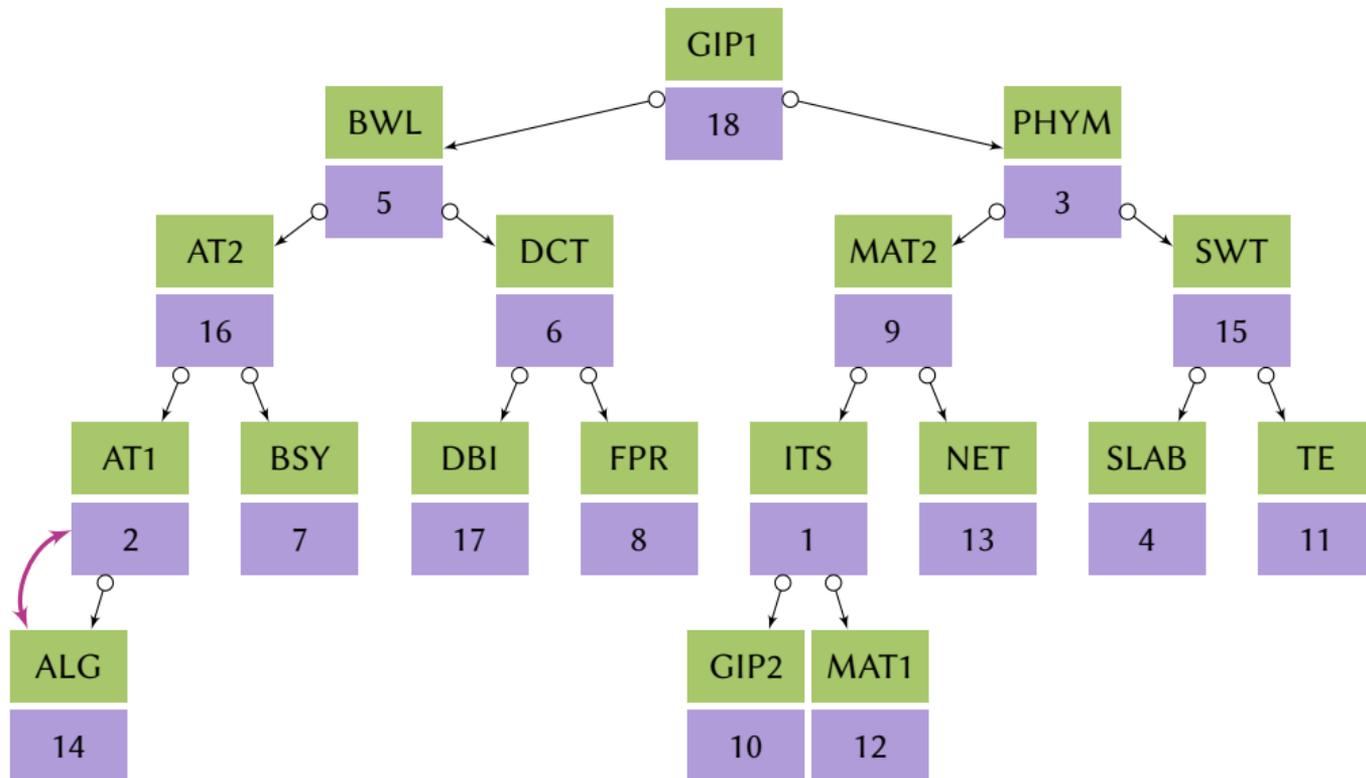
Rotationen

Knoten bewegen, ohne die Sortierung zu ändern

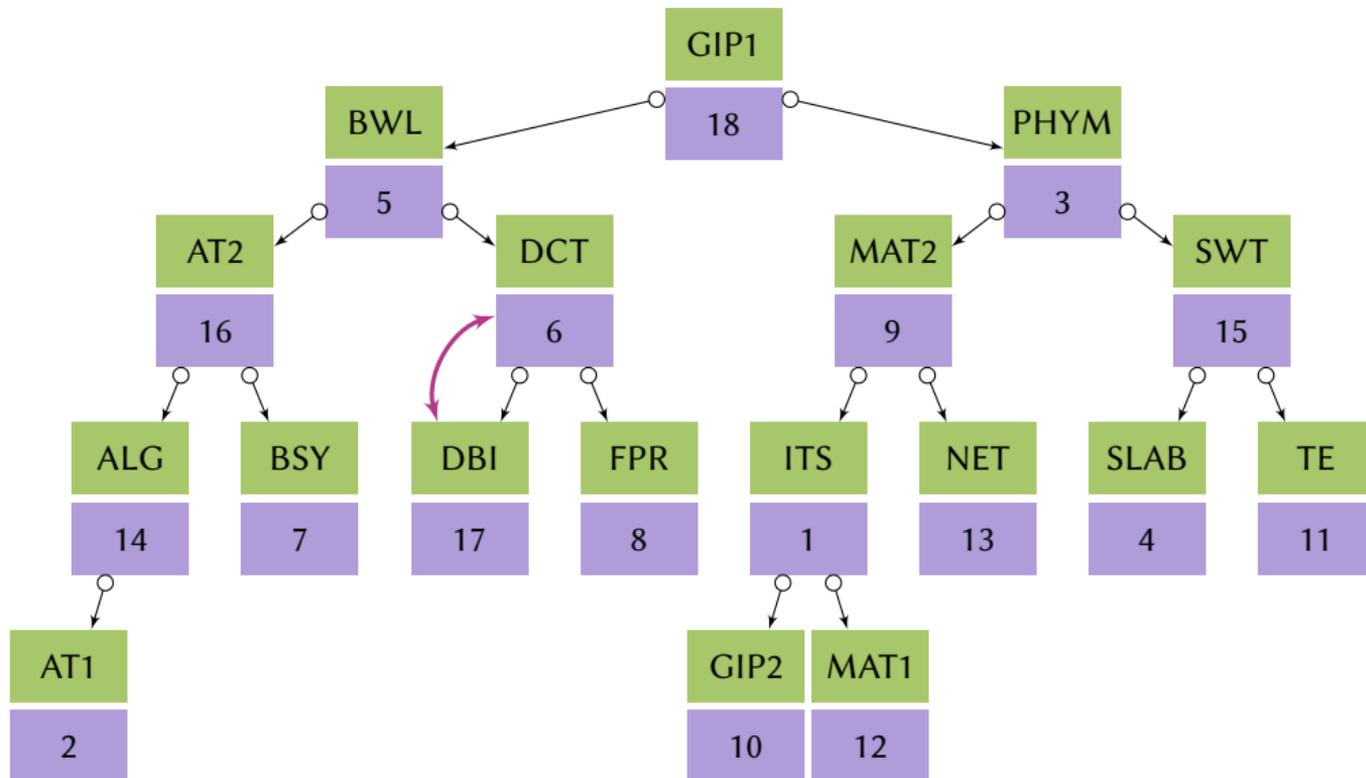
Linksrotation



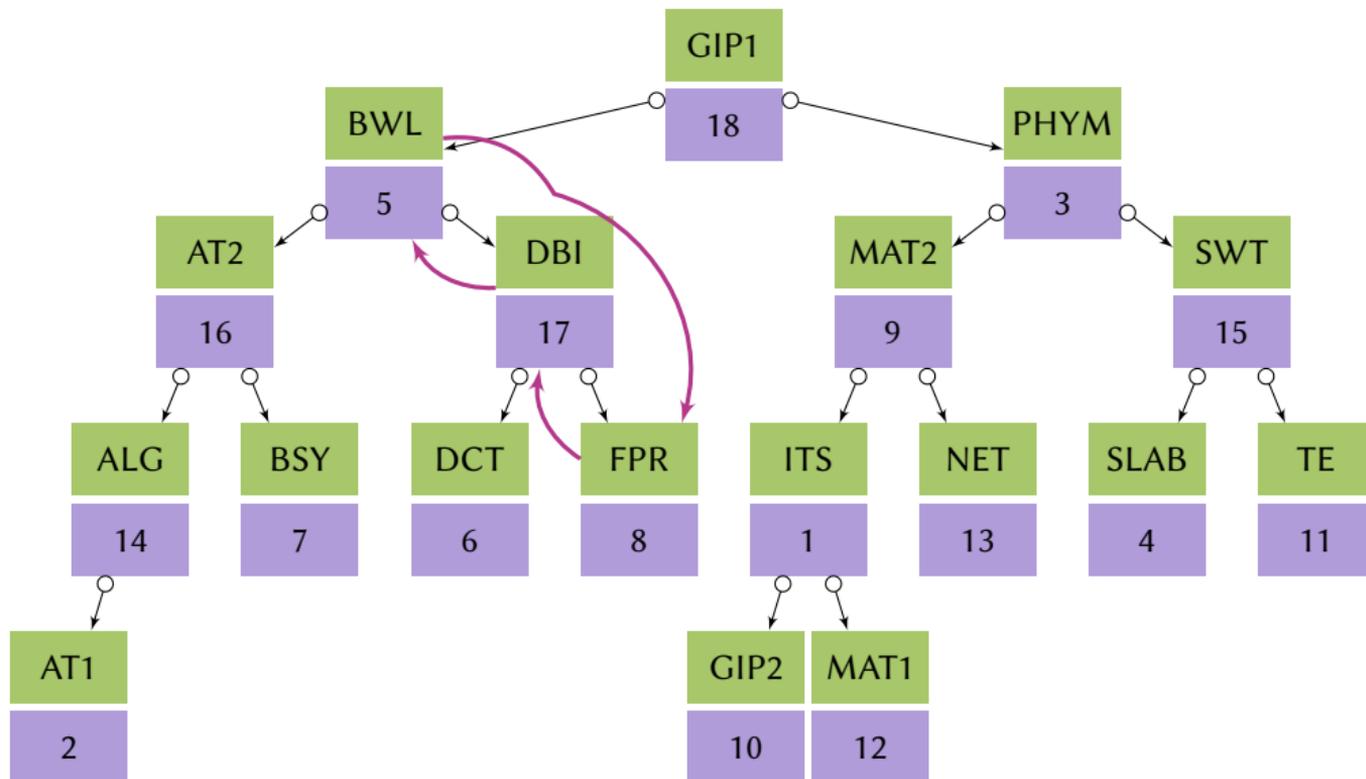
Heap – Bedingung wiederherstellen



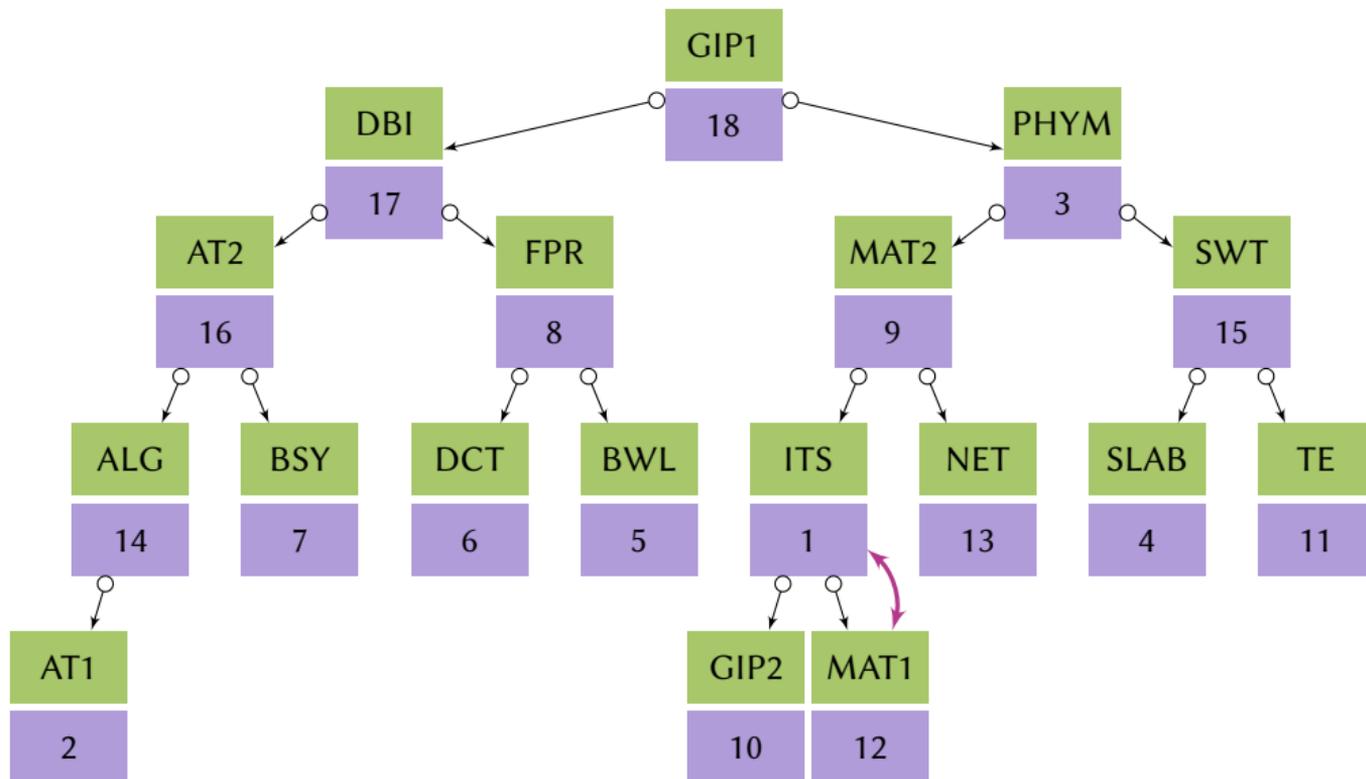
Heap – Bedingung wiederherstellen



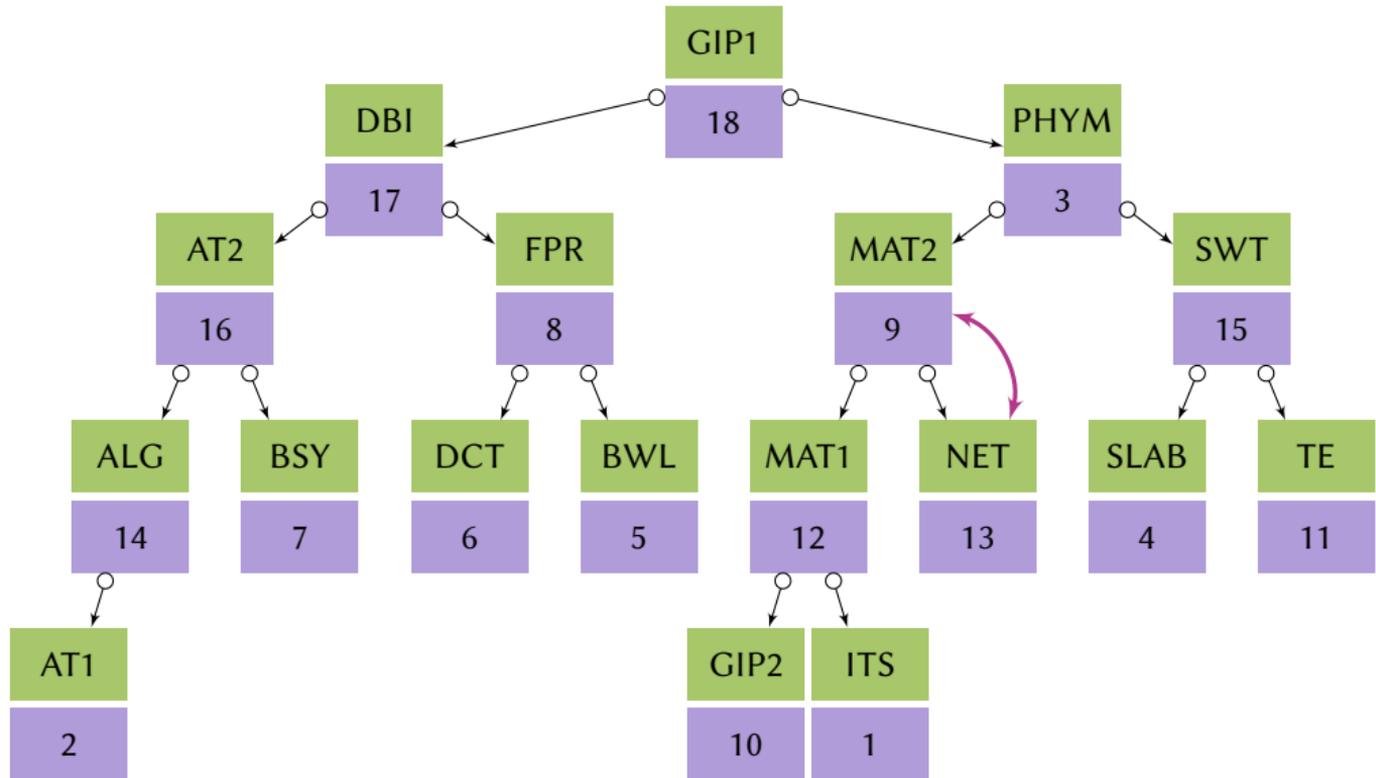
Heap – Bedingung wiederherstellen



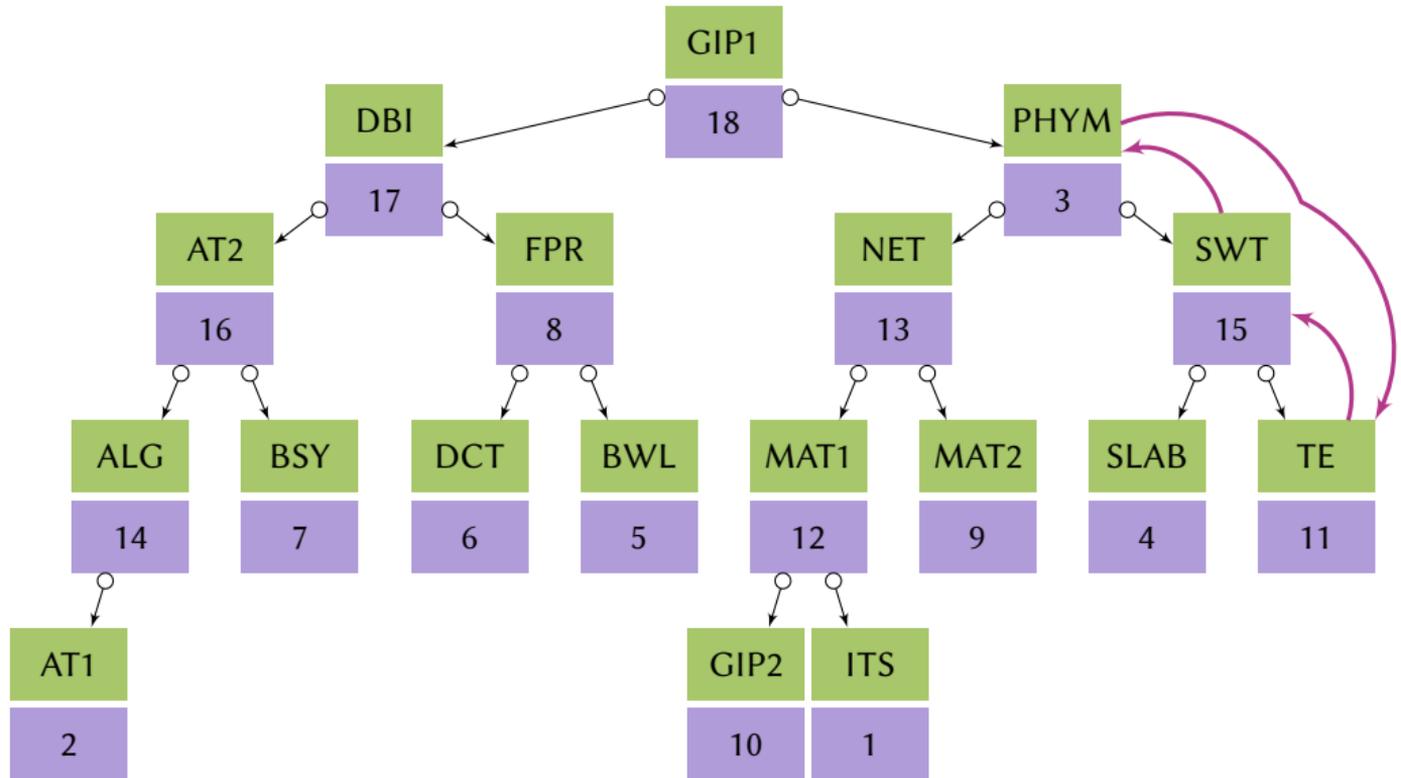
Heap – Bedingung wiederherstellen



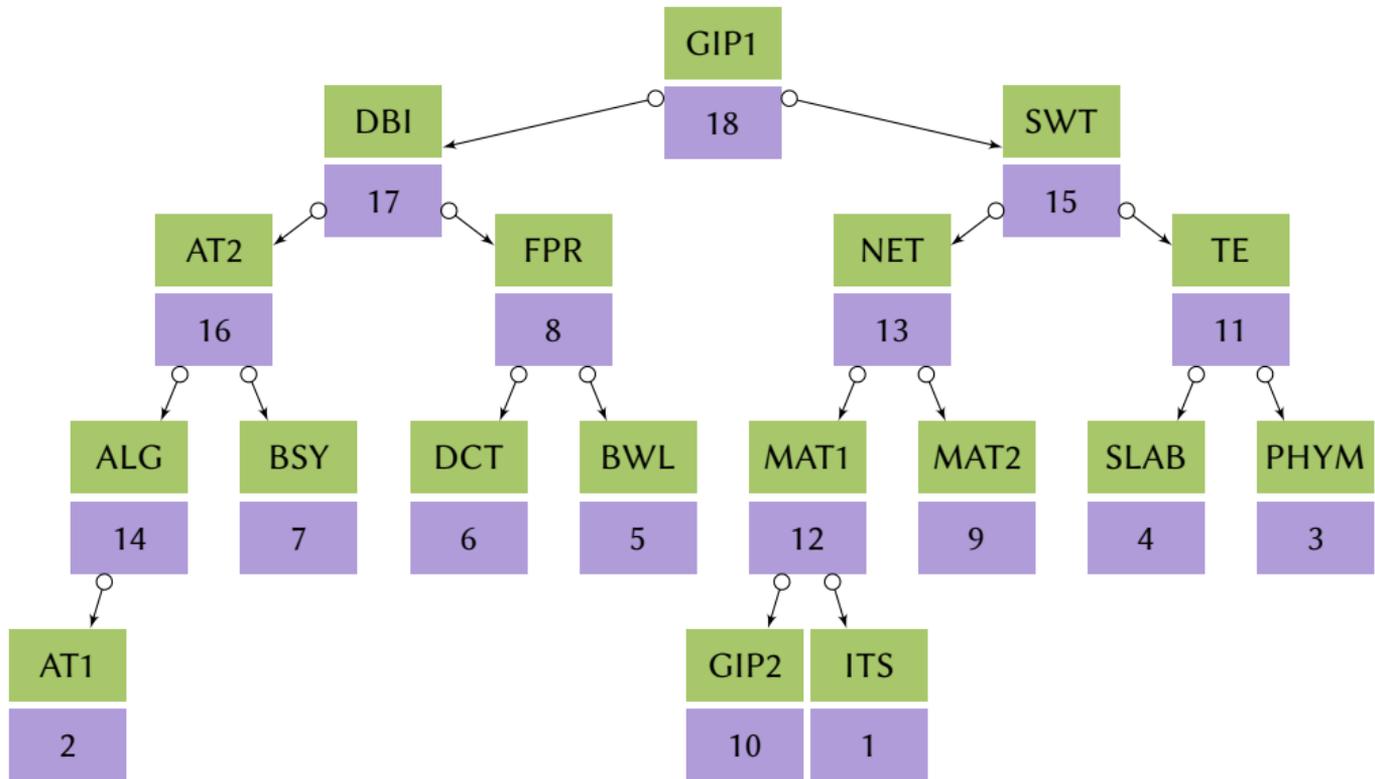
Heap – Bedingung wiederherstellen



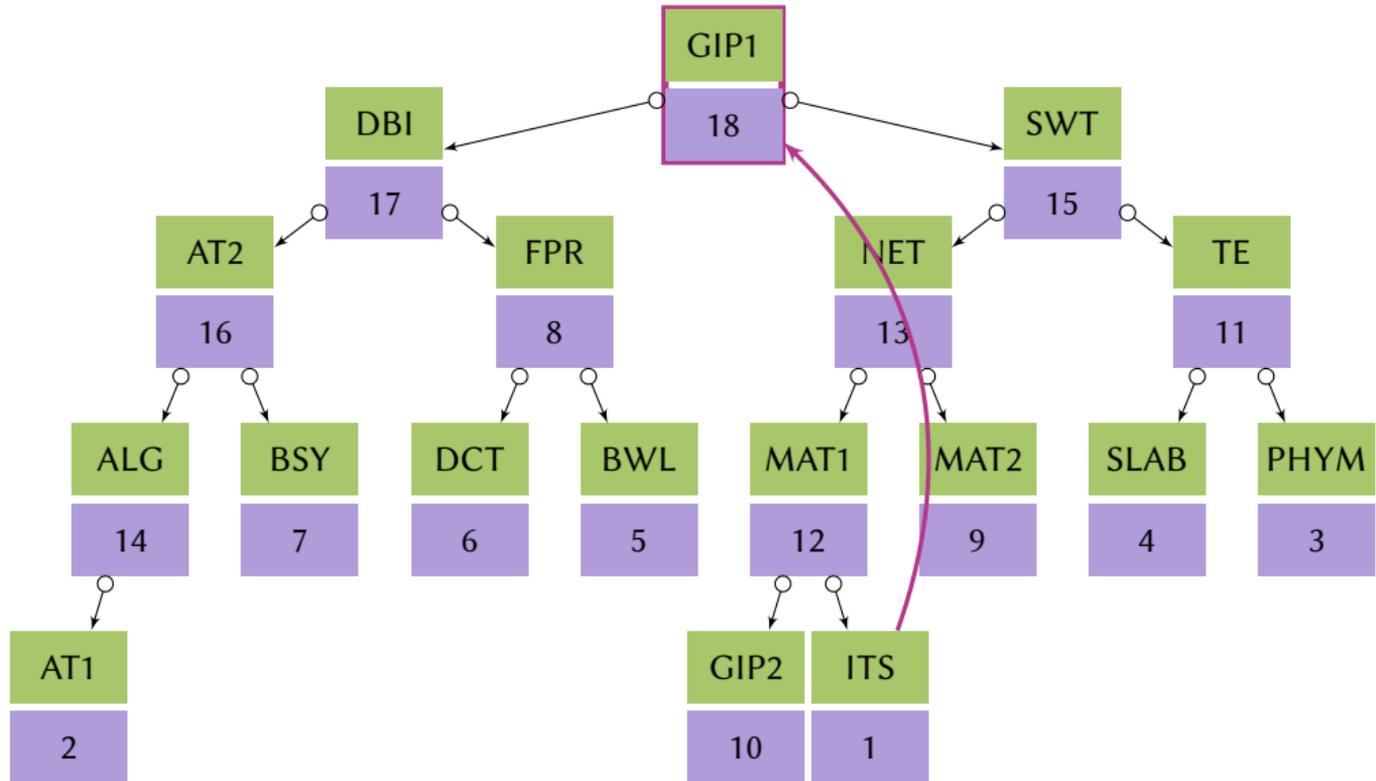
Heap – Bedingung wiederherstellen



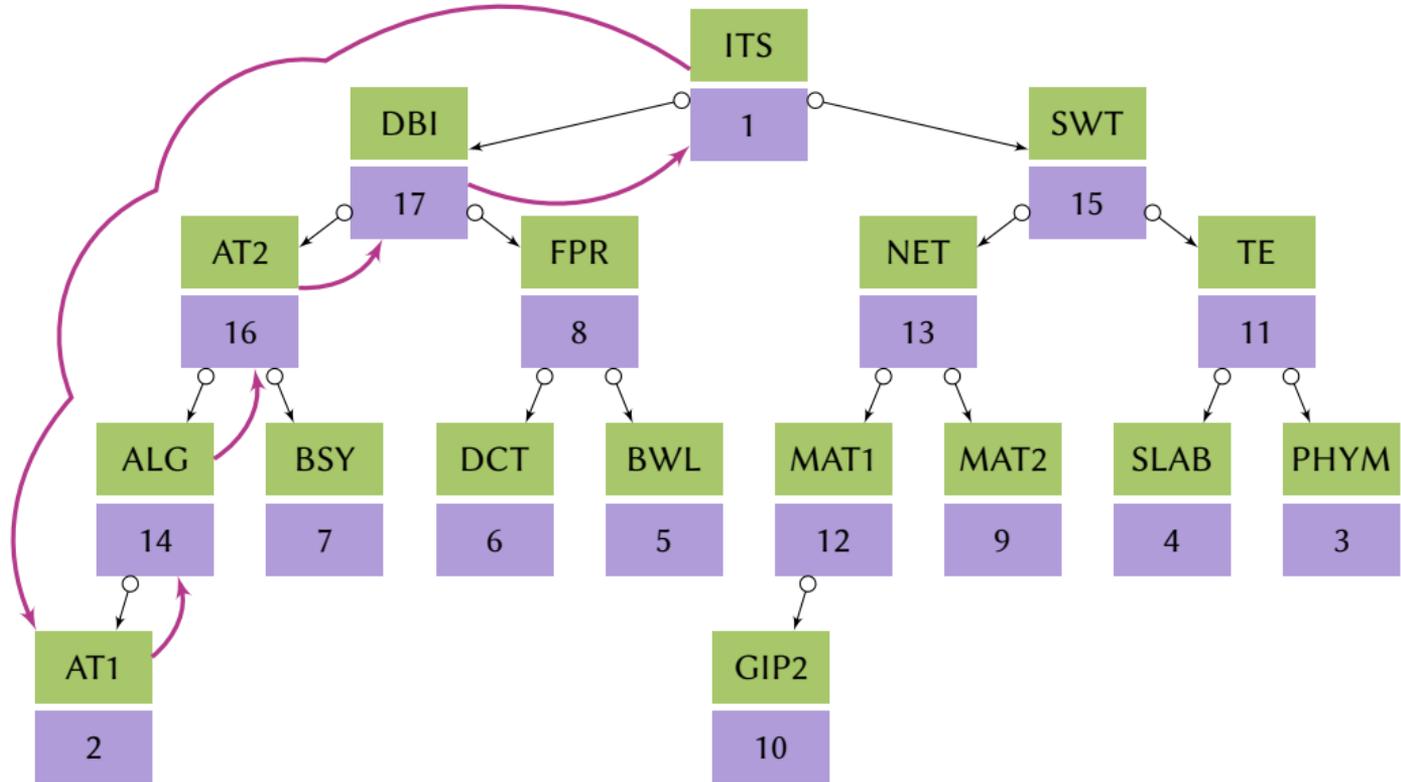
Heap



Heap als Priority-Queue



Heap als Priority-Queue



Definition: Heap

Ein **Heap** ist ein Binärbaum, der die **Heap-Bedingung** erfüllt: Der Key eines Knotens ist

- ▶ größer als oder gleich (Max-Heap)
- ▶ kleiner als oder gleich (Min-Heap)

die/den Keys aller Nachfolger.

Wiederherstellen der Heap-Bedingung

- ▶ Bedingung an genau einer Stelle verletzt: Tausche den betreffenden Knoten solange mit seinem jeweils größtem Nachfolger nach unten, bis die Bedingung erfüllt ist.
- ▶ Bedingung an mehreren Stellen verletzt: Obige Schritte während Post-Order Traversierung.

Array-Repräsentation eines Binärbaums

