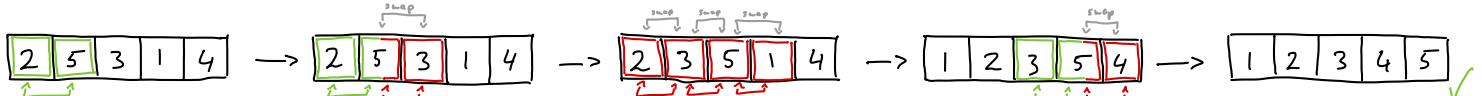


## Insertion Sort



Da  $2 < 5$  muss nicht getauscht werden.

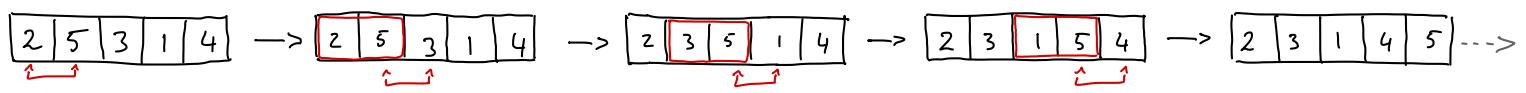
Da  $5 > 3$ , wird getauscht, bis  $3 >$  als zu vergleichendes Element. In dem Fall wird 3 nur mit 5 getauscht.

1 wird bis an die erste Stelle getauscht.

Da  $5 > 4$  aber  $3 < 4$  wird die 4 nur mit der 5 getauscht.

Das Array ist sortiert.

## Bubble Sort



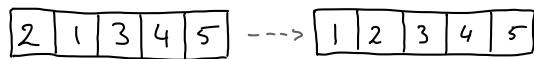
Mit dem ersten Element beginnend vergleichen.  $2 < 5$ , also nicht tauschen.

zweites und drittes Element vergleichen. Da  $5 > 3$ , tauschen.

$5 > 1 \rightarrow$  tauschen

$5 > 4 \rightarrow$  tauschen

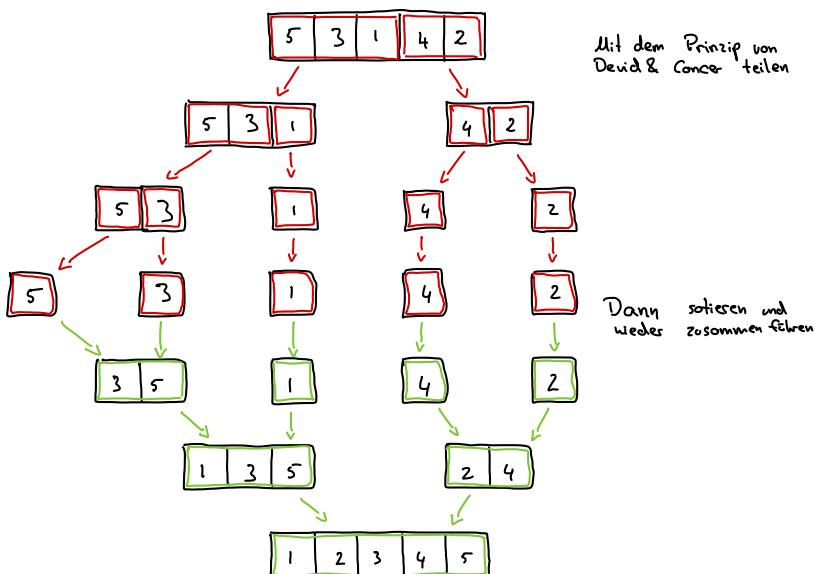
Ergebnis der ersten Iteration



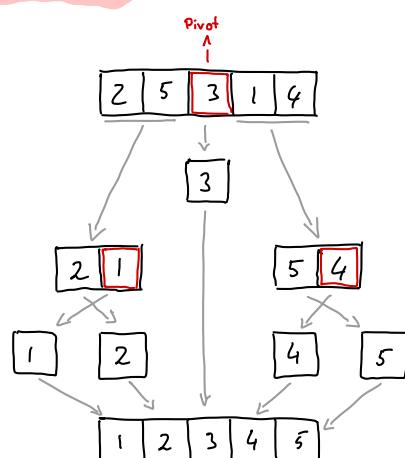
Ergebnis der zweiten Iteration.

Ergebnis der dritten Iteration.

## Merge Sort



## Quicksort

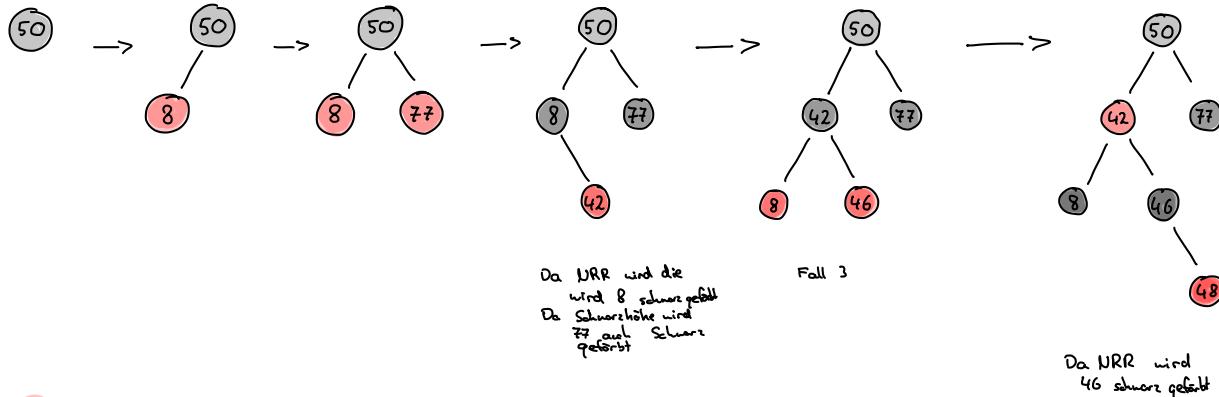


Ähnlich wie im Merge Sort teilen und sortieren. Es wird ein Pivot Element gewählt, und anschließend von diesem werden die Elemente dann in zwei Teilarrays sortiert. Danach wird Quicksort rekursiv erneut aufgerufen.

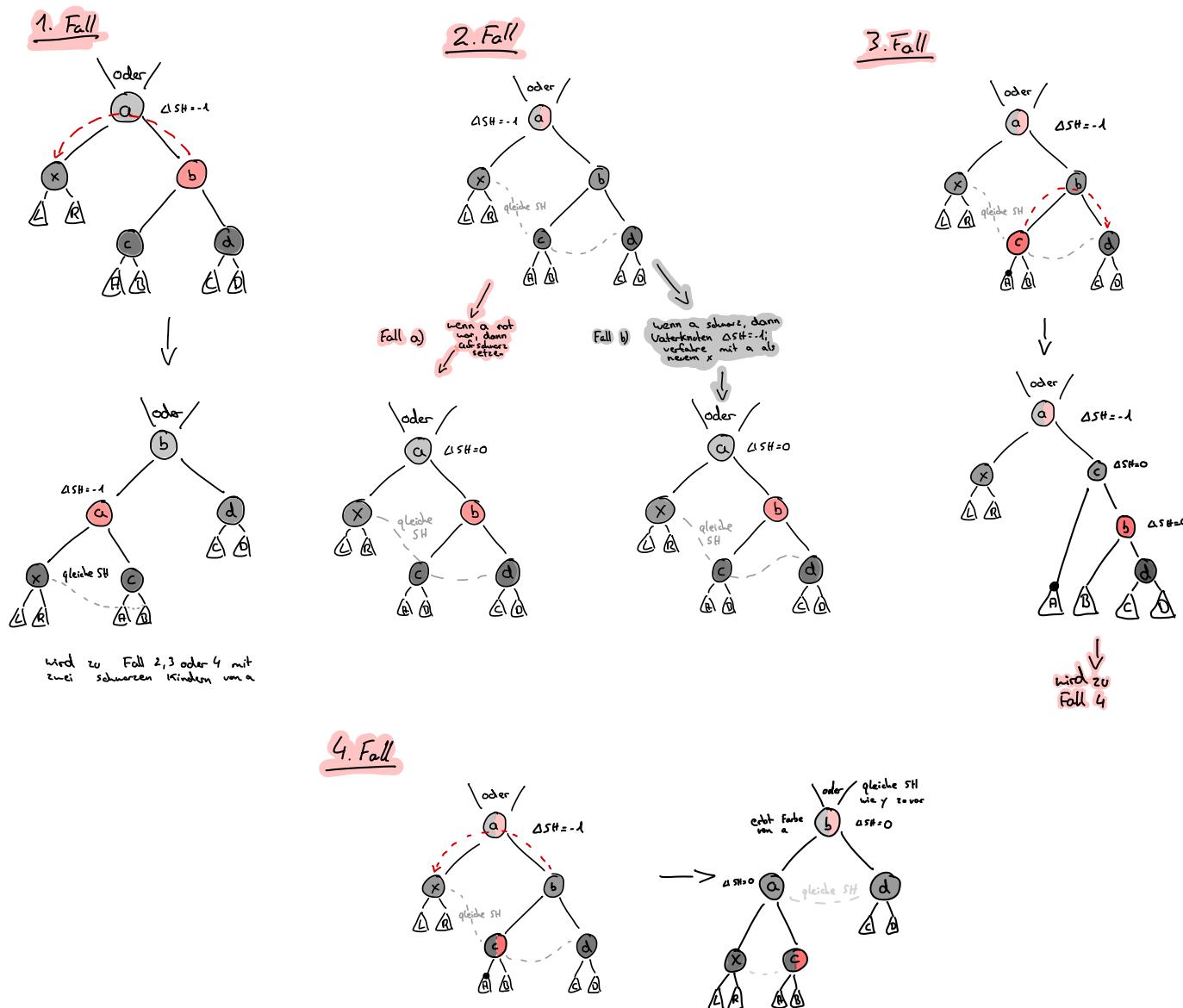
# Rot - Schwarz Bäume

Bsp.: Built-Tree

insert 50, 8, 77, 42, 46, 48

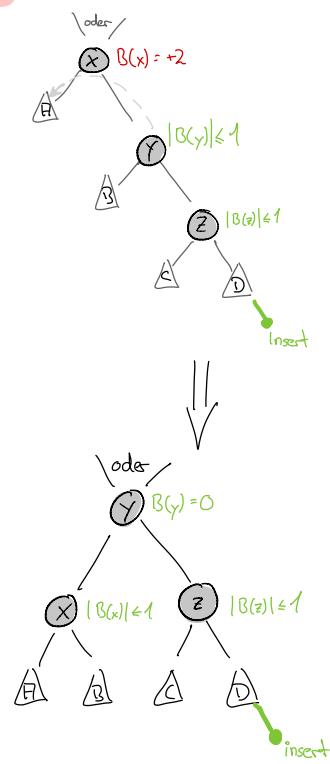


## Löschen Fixup

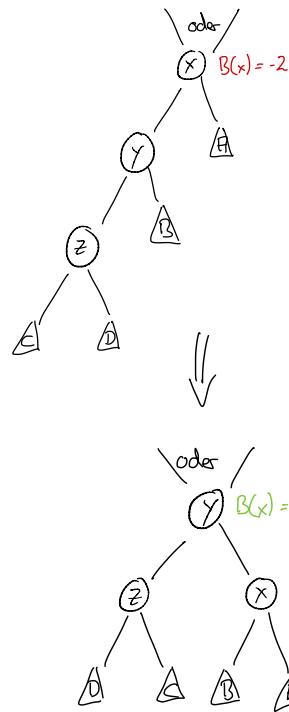


AVL

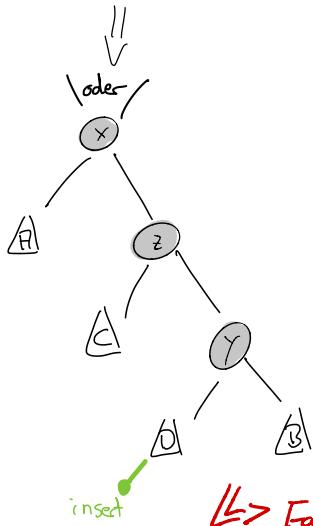
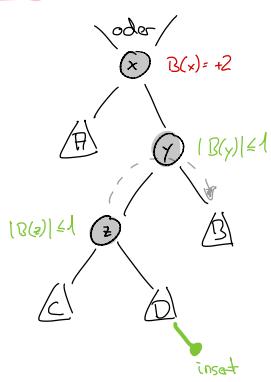
## 1. Fall



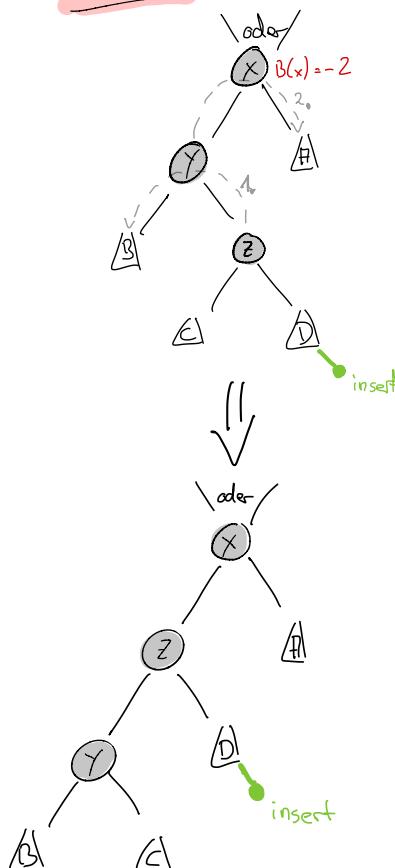
## lb. Fall



## 2. Fall



## 2b. Fall

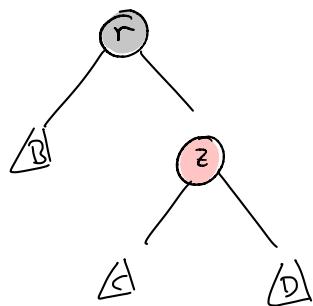


L> Fall 1

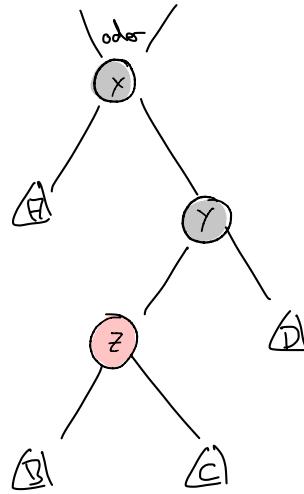
L> Fall 1b

Splay

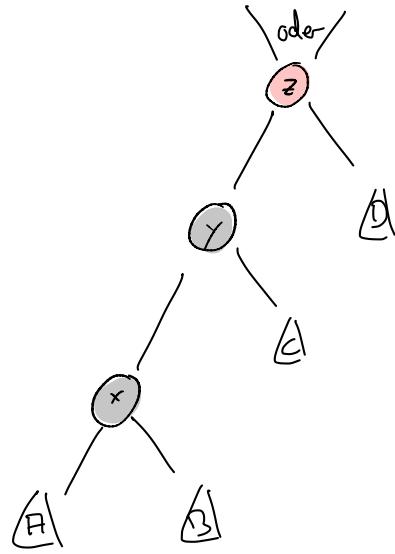
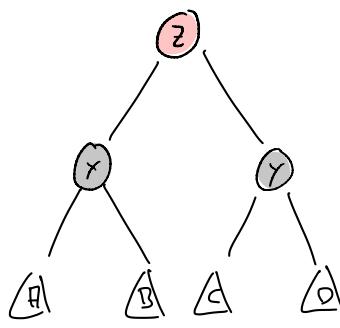
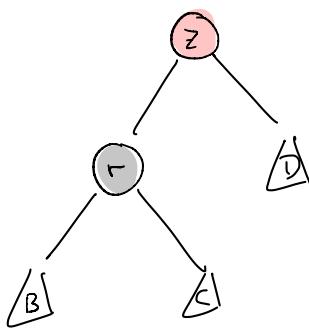
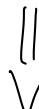
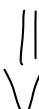
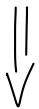
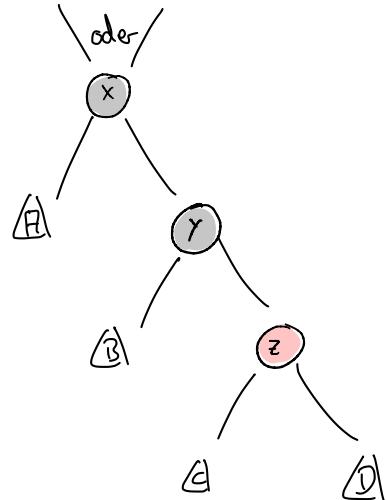
Zig ( $r = \text{word}$ )



Zig-Zag

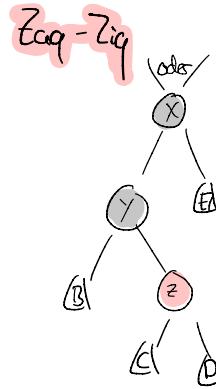
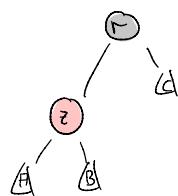


Zig-Zig

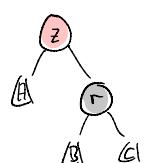
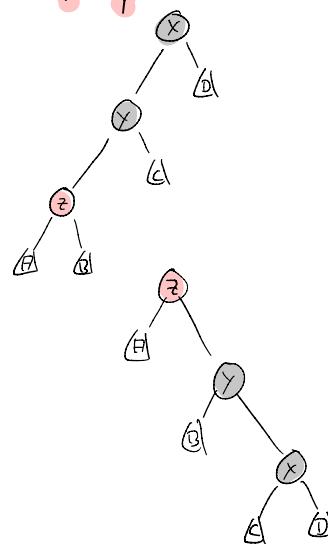


Final zig dazw:

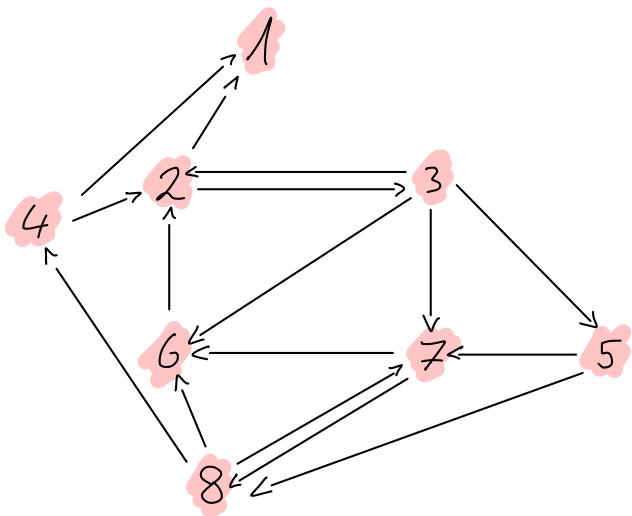
Zag



Zag-Zag



BFS

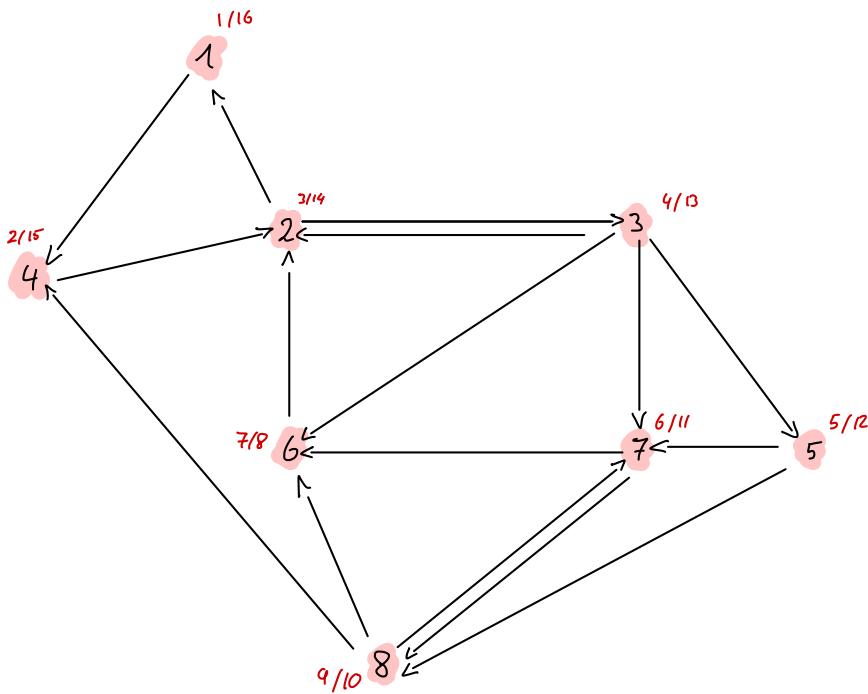


Iteration	U	v	Q
0	-	∅	[7]
1	7	6,8	∅
2	6	2	[8]
3	8	4	[2]
4	2	1,3	[4]
5	4	∅	[1,3]
6	1	∅	[3]
7	3	5	∅
8	5	∅	∅

Adjazenzmatrix

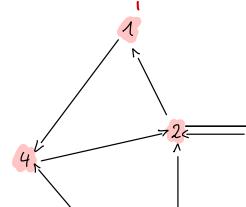
	1	2	3	4	5	6	7	8
1	1	0	0	0	0	0	0	0
2	0	1	1	0	0	0	0	0
3	0	1	0	1	0	0	1	0
4	1	1	0	0	0	0	0	0
5	0	0	0	0	0	1	1	1
6	0	1	0	0	0	0	0	0
7	0	0	0	0	0	1	0	1
8	0	0	0	1	0	1	1	0

## DFS



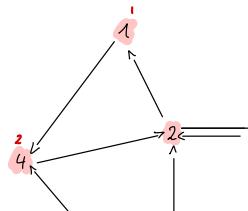
1

Mit dem Start Knoten beginnend, setzt man diesem den Time-Stamp 1.



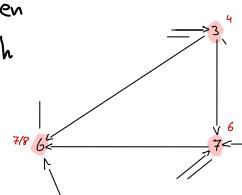
2

Dem nächsten Knoten, den man "entdeckt" gibt man den Time-Stamp 2. Hat man mehrere zur Auswahl, beginnt man mit dem, der den niedrigsten Wert hat.



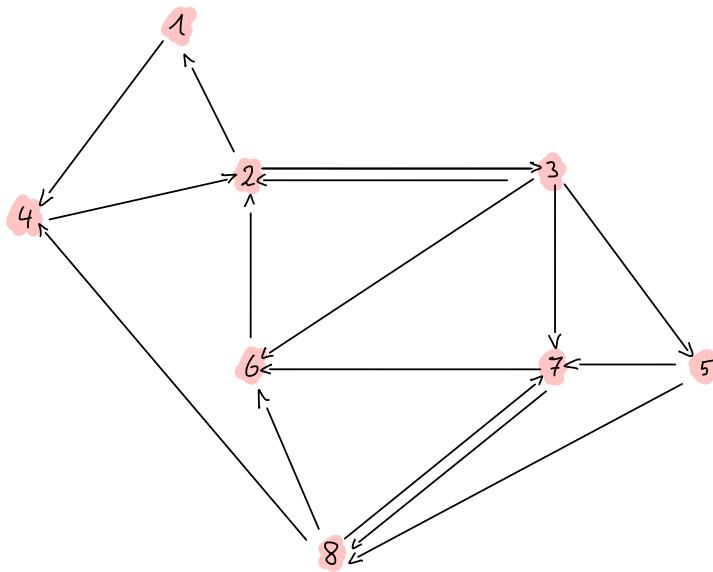
3

Gibt es von einem Knoten keinen neuen zu entdecken, erhält dieser den finish Time-Stamp, in dem Fall den Knoten 6 den visit Time-Stamp 7 und den finish Time Stamp 8. Man geht zurück zu dem Knoten, von dem man gekommen ist.

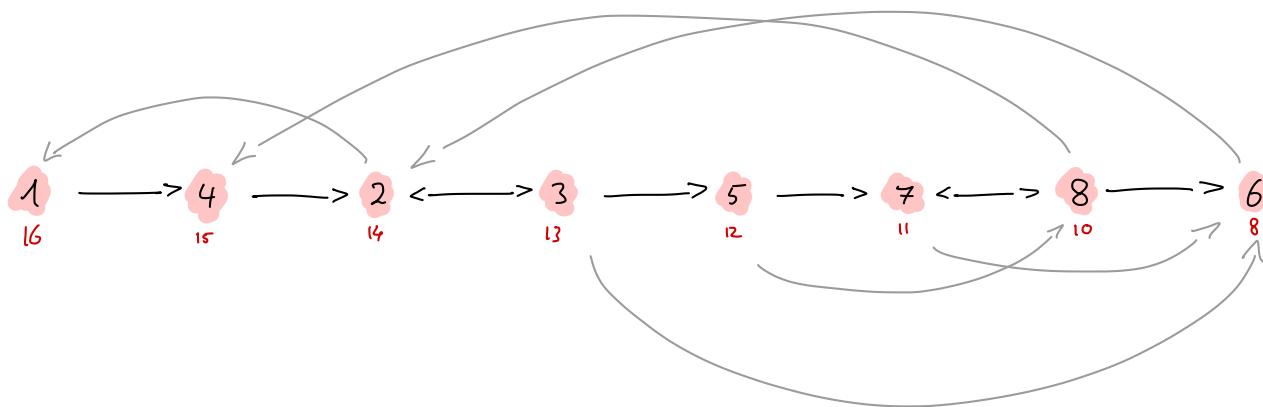


## TOPO-Sort

Mit DFS



Von links beginnend alle Kinder mit dem höchsten Time-Stemp aufzeichnen und Knoten einzeichnen. Für TOPO-Sort dürfen nur Knoten von links nach rechts laufen.



1. Knoten mit null Vorgängen bestimmen
2. Einen Knoten wählen
3. diesen von 0 auf -1 setzen
4. Bei allen Nachfolgern einen abziehen

Knoten	Vorgänger
1	0
2	3
3	1
4	2
5	1
6	3
7	3
8	2

Bei dem gegebenen Graphen ist keine topologische Sortierung möglich