



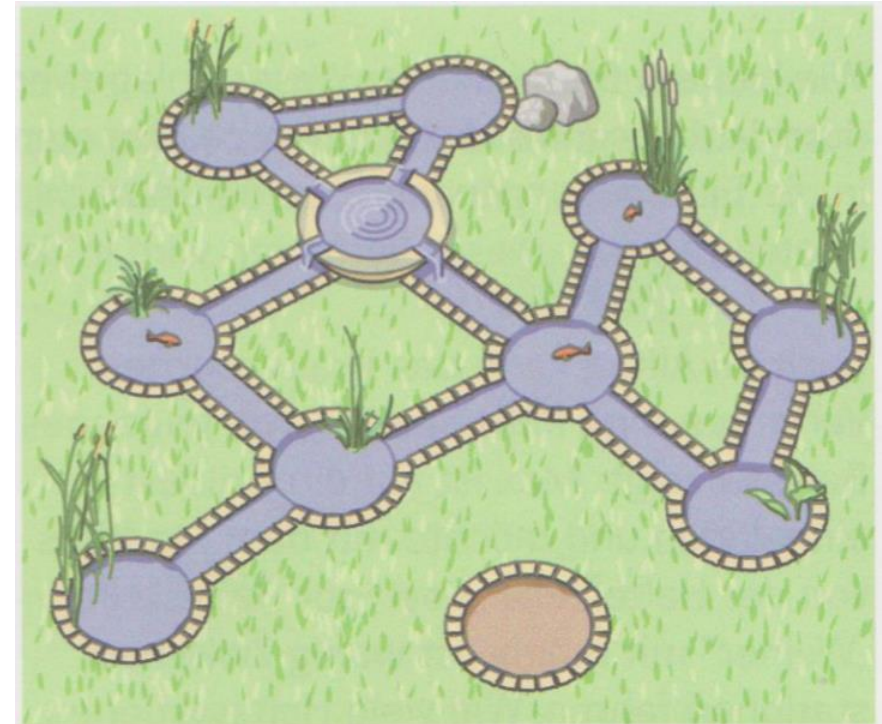
# Graphendurchlauf II – Breitensuche

## Der Weg ist das Ziel



# Problemstellung

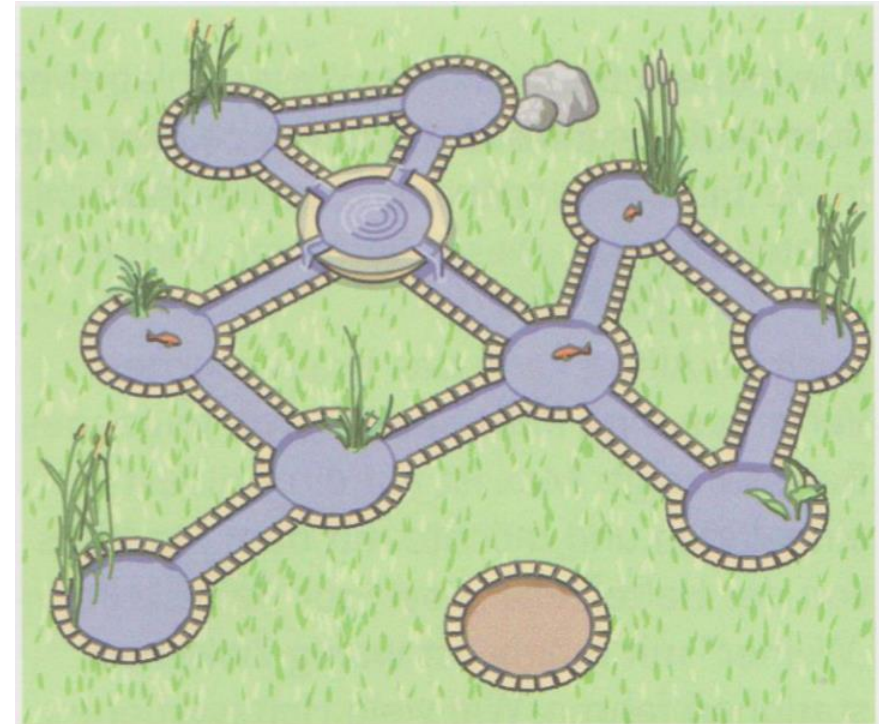
- Herr Reich plant für seinen Garten den Bau einer Wasseranlage.
- Von der Quelle aus werden alle Wasserbecken, die durch gleich lange Wasserrinnen miteinander verbunden sind, mit Wasser versorgt.





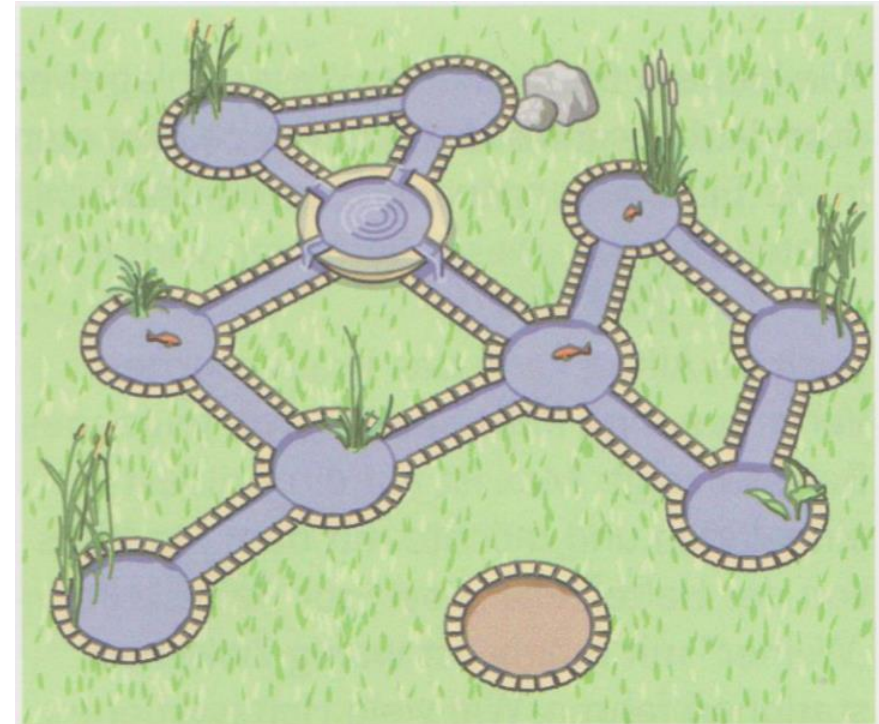
# Problemstellung

- Annahme:
  - Alle Verbindungsrippen sind gleich lang.
  - Das Wasser benötigt für den Durchlauf von einem Becken zum nächsten je 1 Minute.
  - Das Auffüllen der Becken fällt ohne Zeit ins Gewicht.



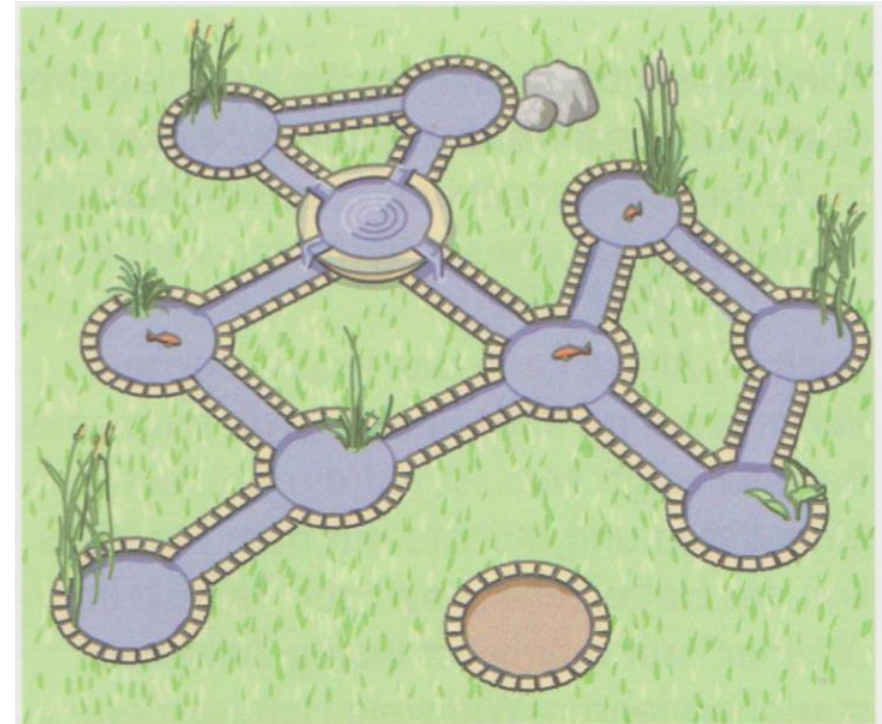
# Problemstellung

- Annahme:
  - Alle Verbindungsrinnen sind gleich lang.
  - Das Wasser benötigt für den Durchlauf von einem Becken zum nächsten je 1 Minute.
  - Das Auffüllen der Becken fällt ohne Zeit ins Gewicht.
- Zu welchen Zeiten erreicht das Wasser die einzelnen Becken?

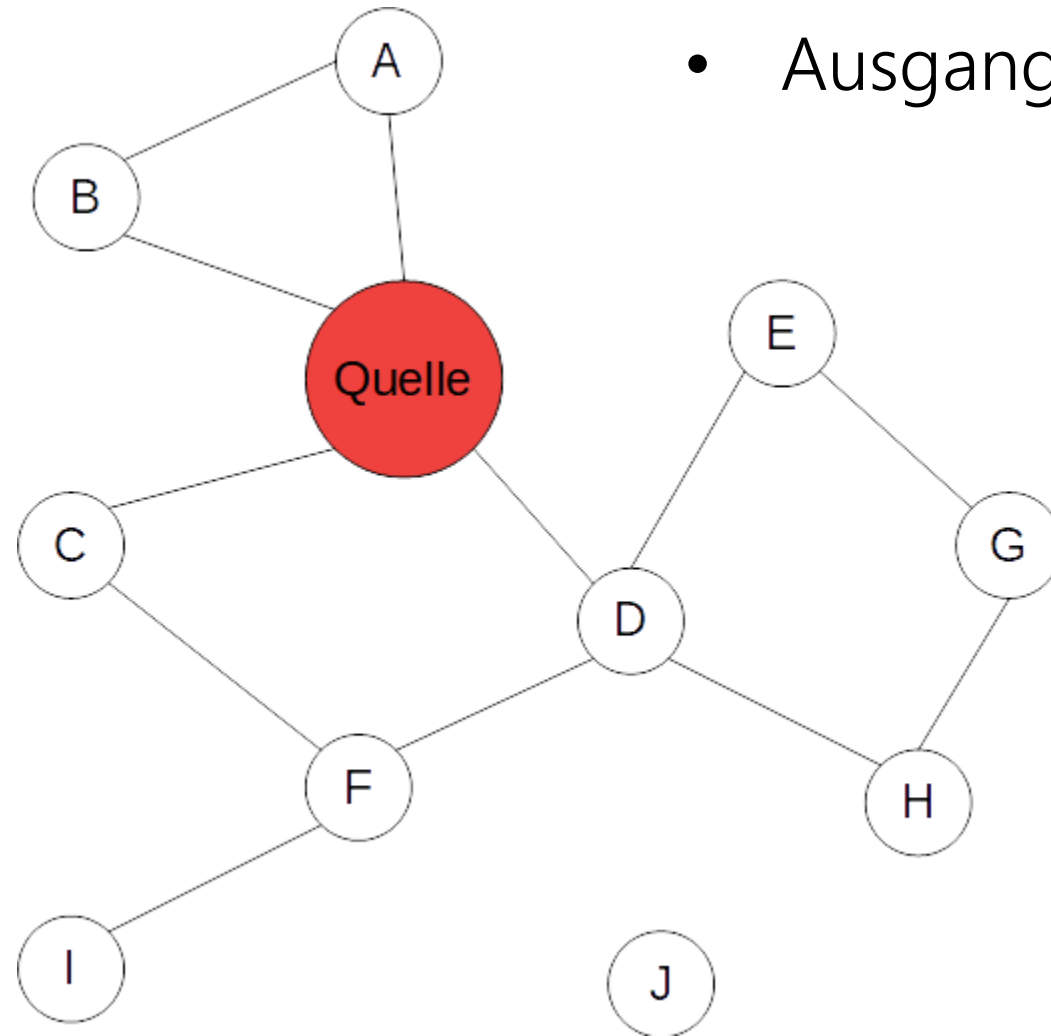


# Problemstellung

- Wie könnte man das Szenario in einem Graphen darstellen?
- Beschreibt den Graphen und den Algorithmus.

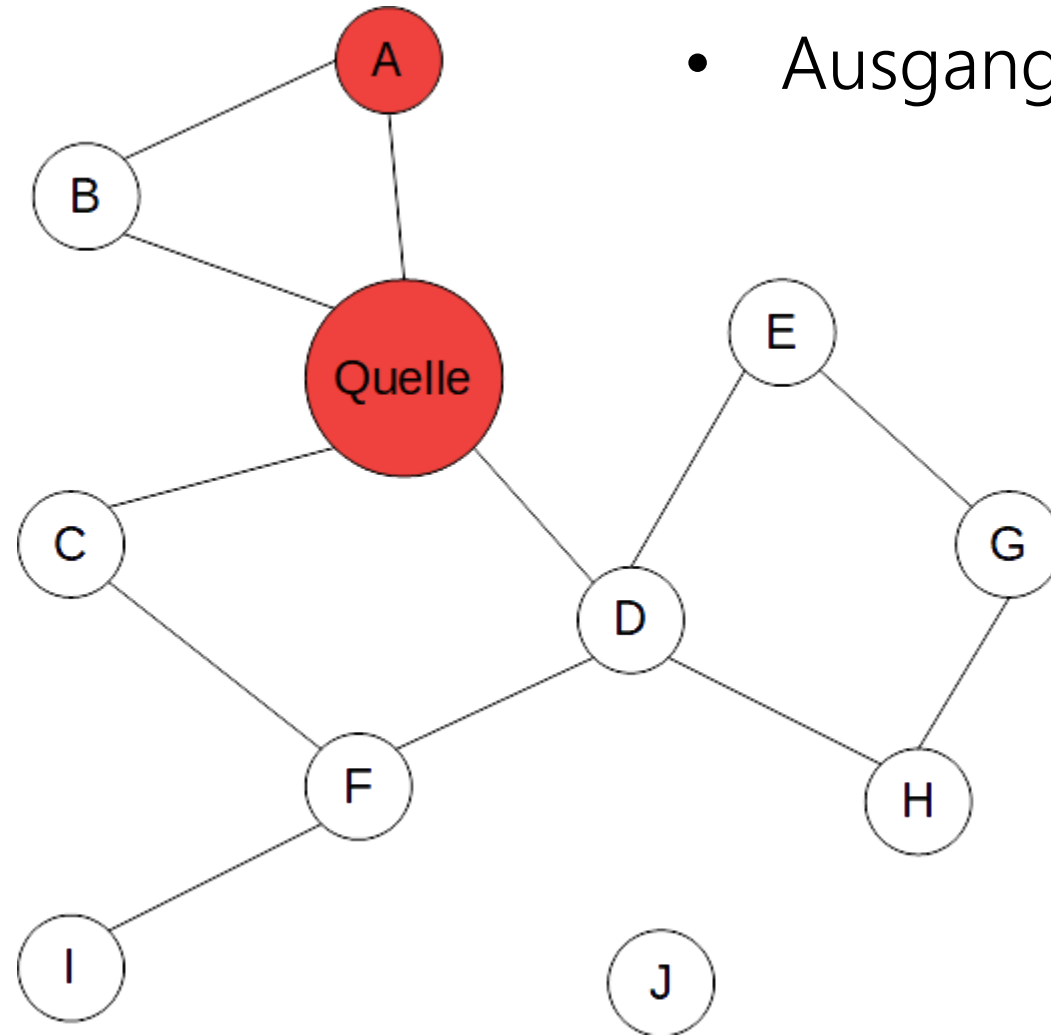


# Breitensuche



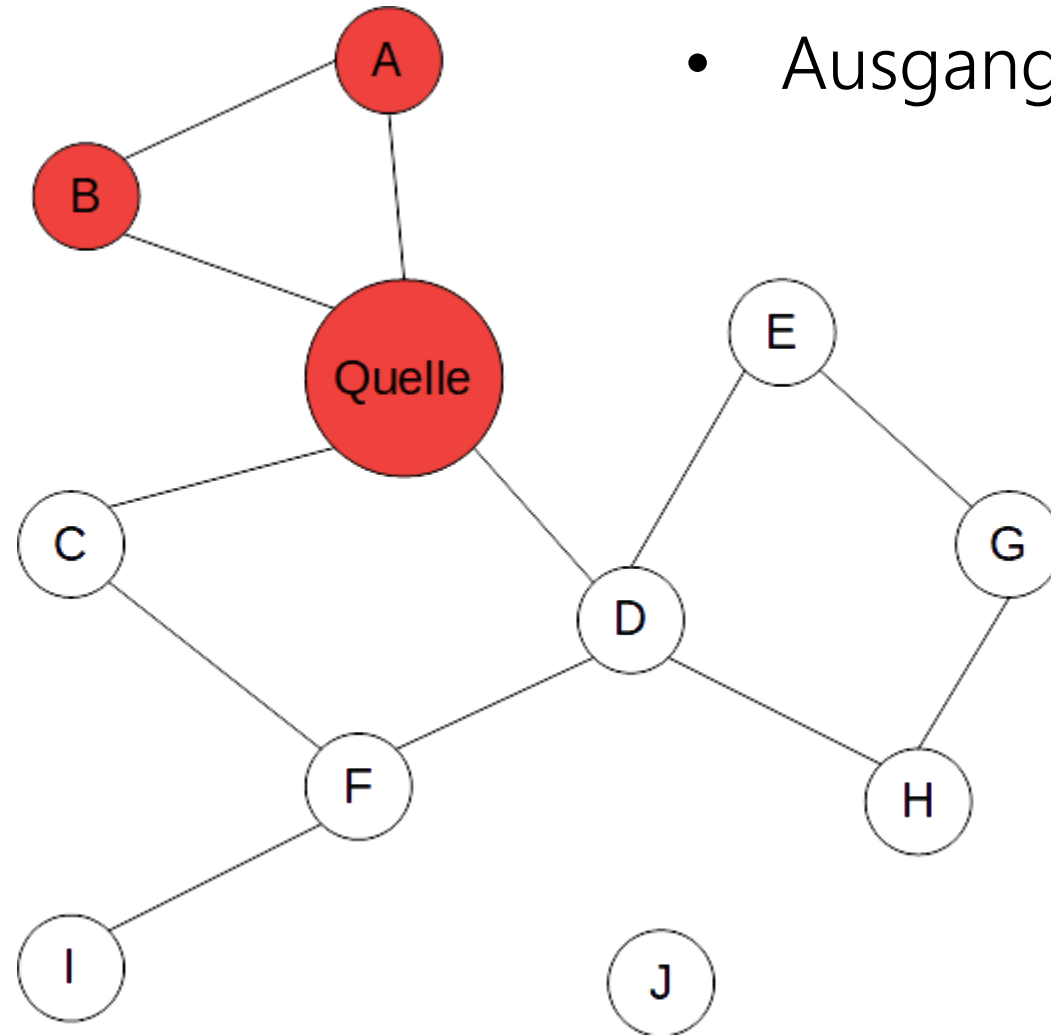
- Ausgangsknoten: Quelle

# Breitensuche



- Ausgangsknoten: Quelle

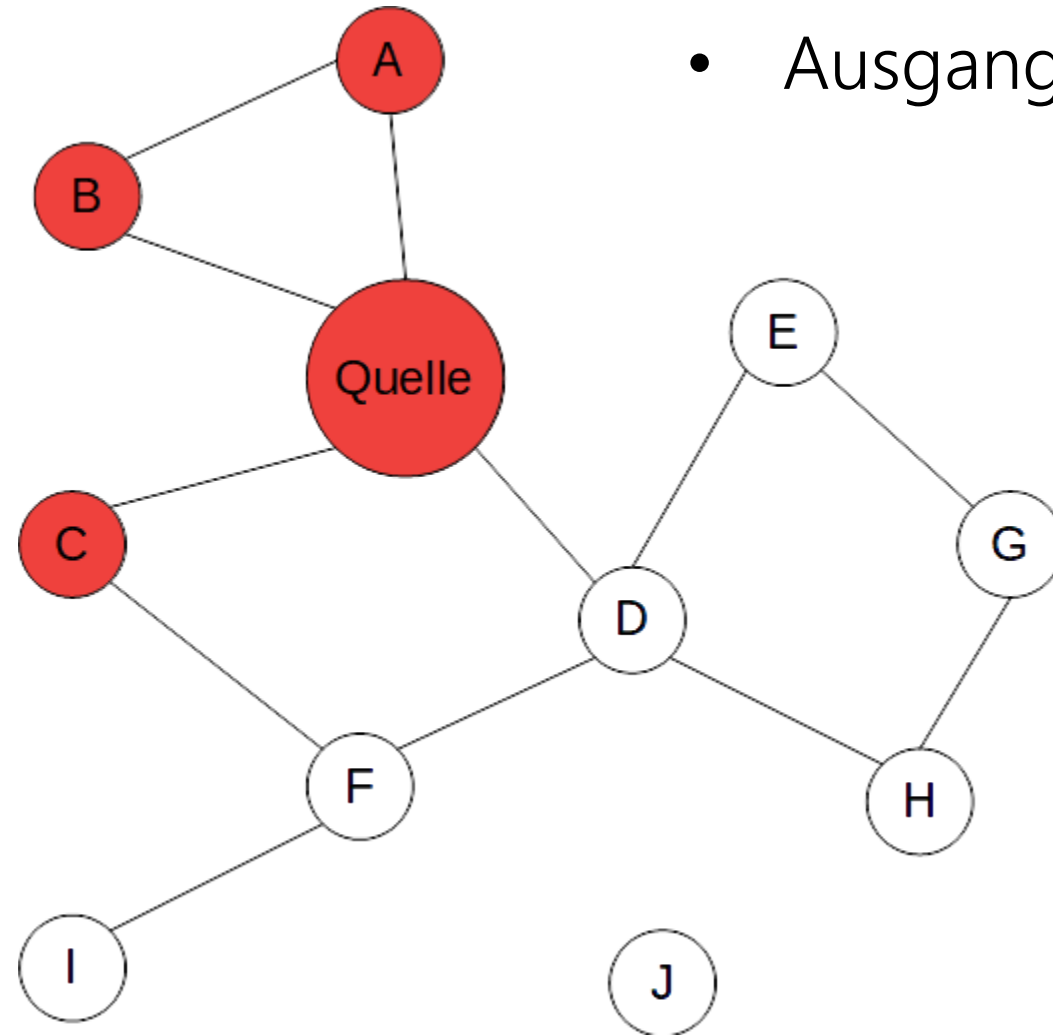
# Breitensuche



- Ausgangsknoten: Quelle

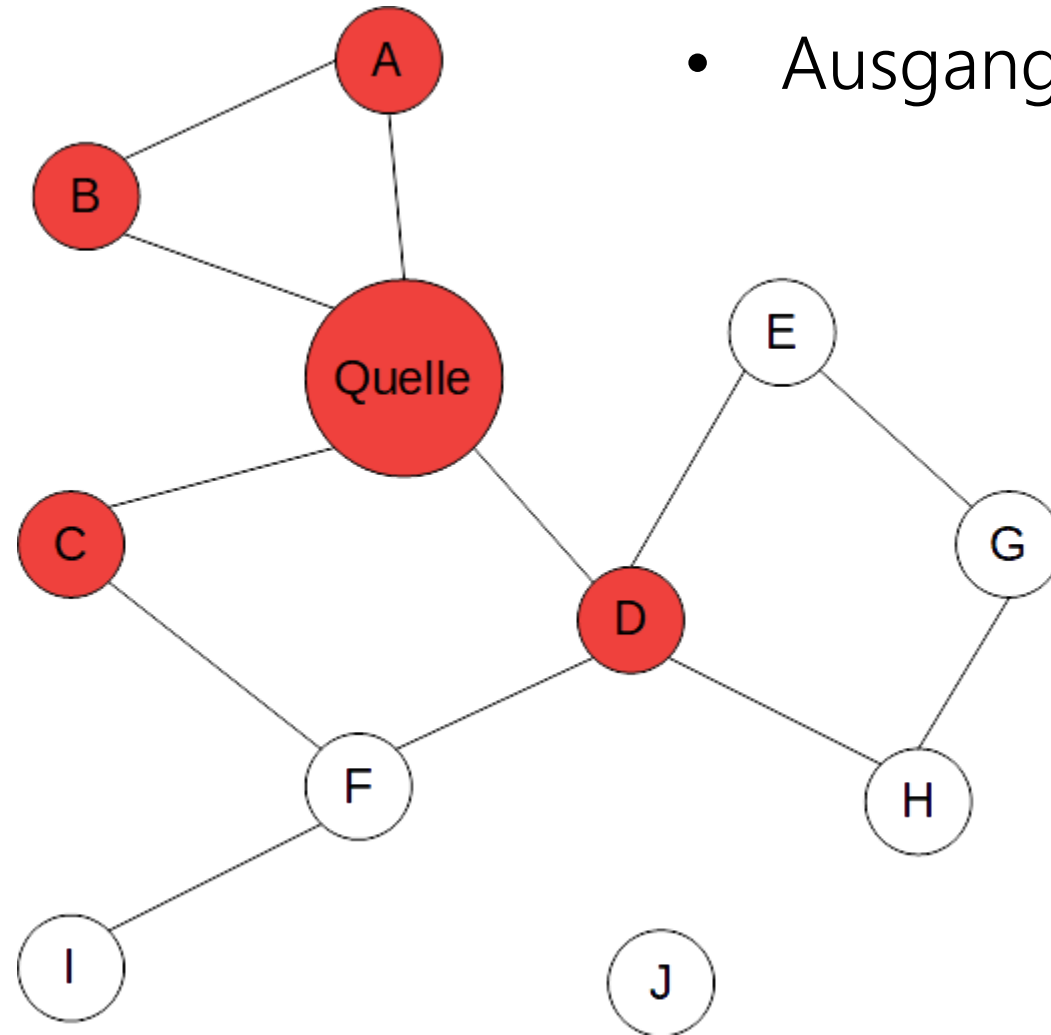


# Breitensuche



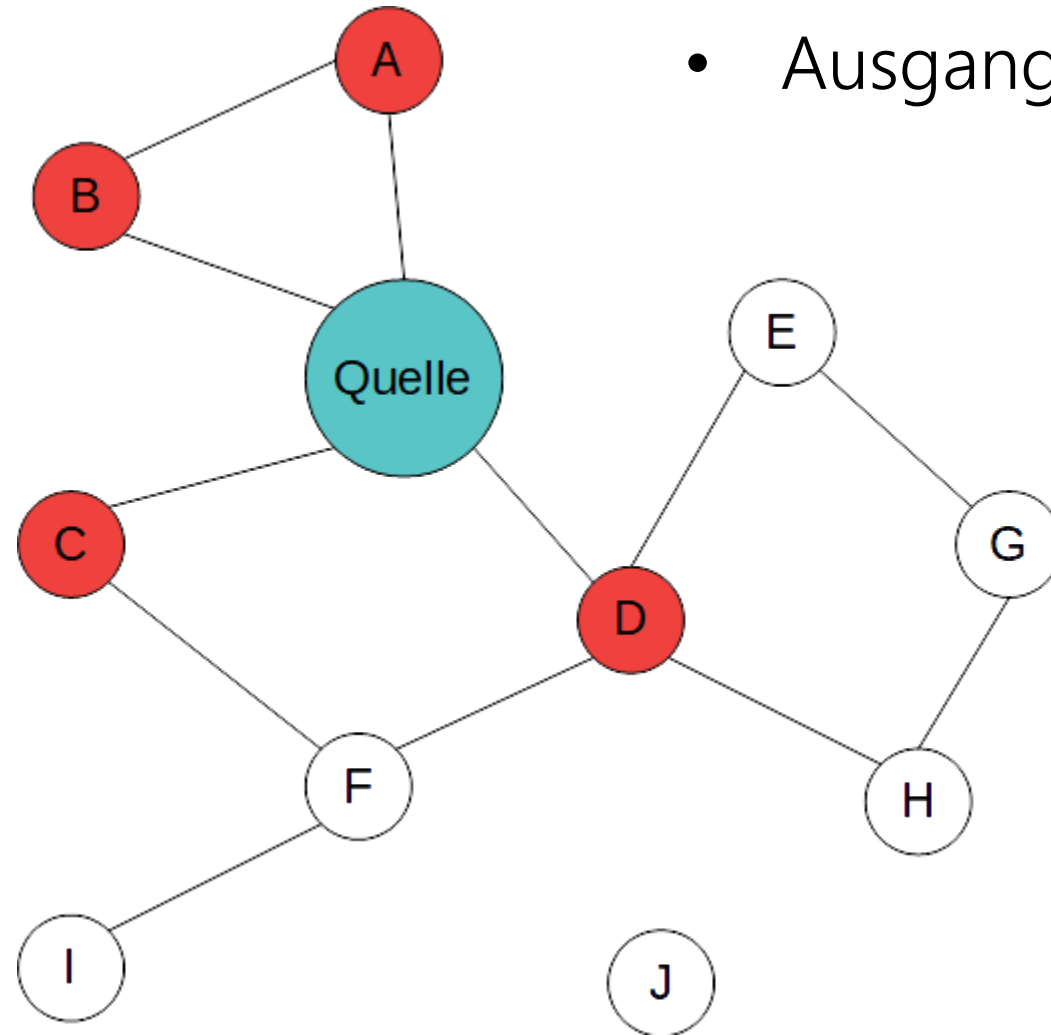
- Ausgangsknoten: Quelle

# Breitensuche



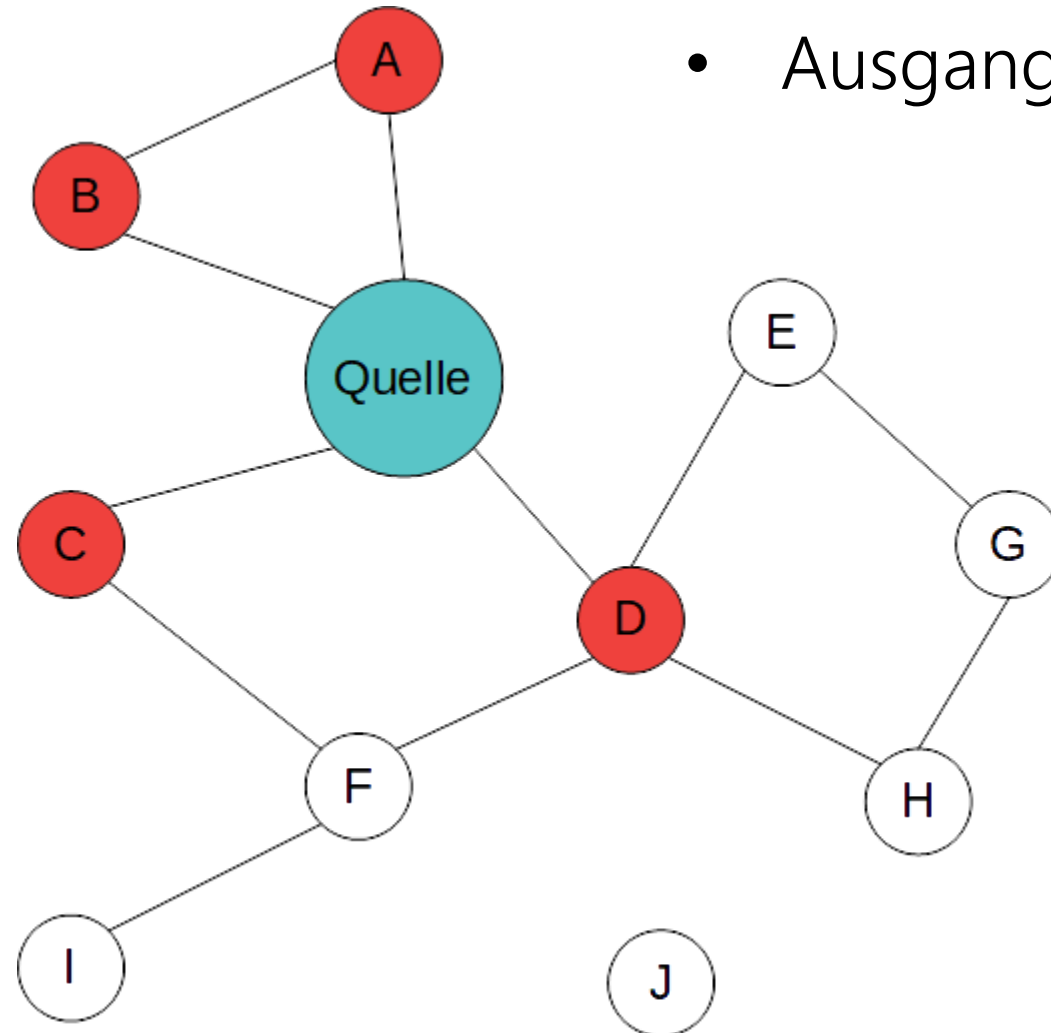
- Ausgangsknoten: Quelle

# Breitensuche



- Ausgangsknoten: Quelle

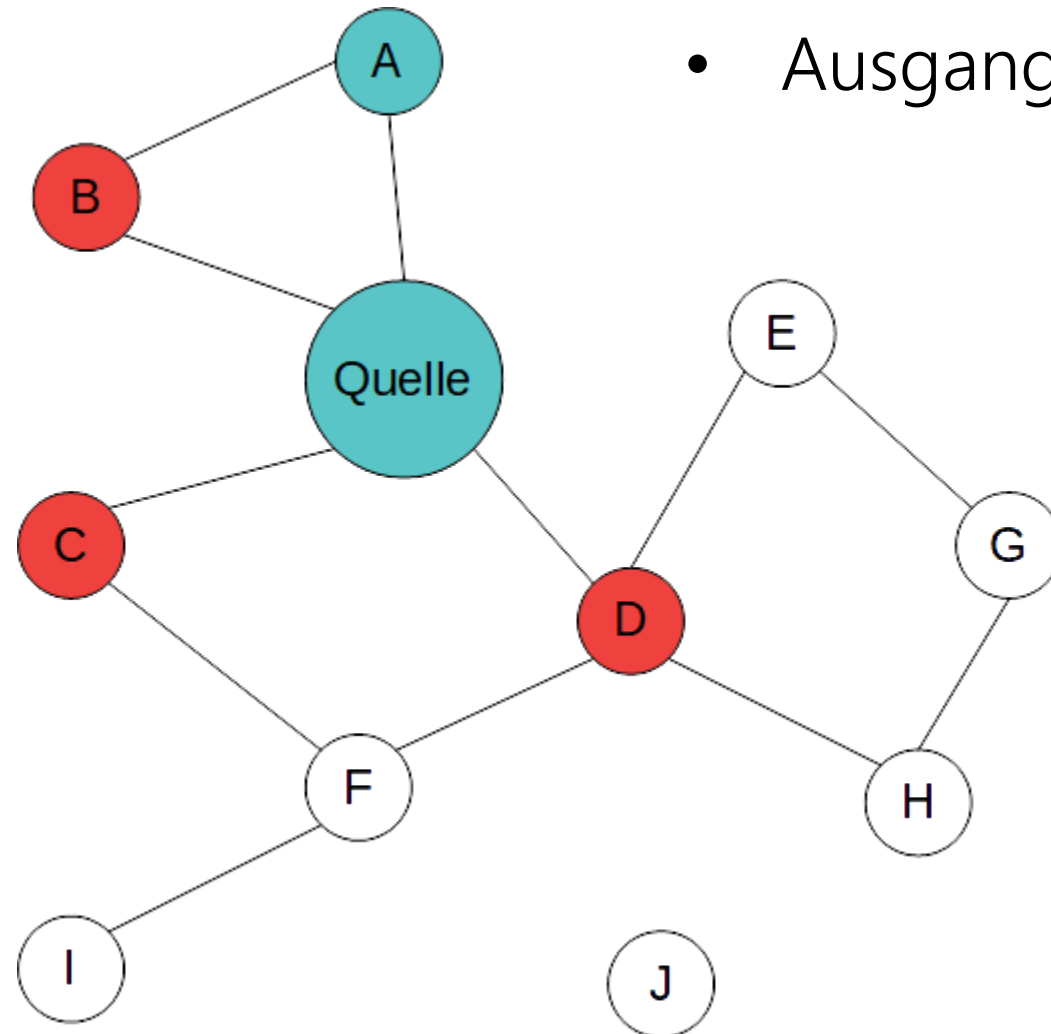
# Breitensuche



- Ausgangsknoten: A

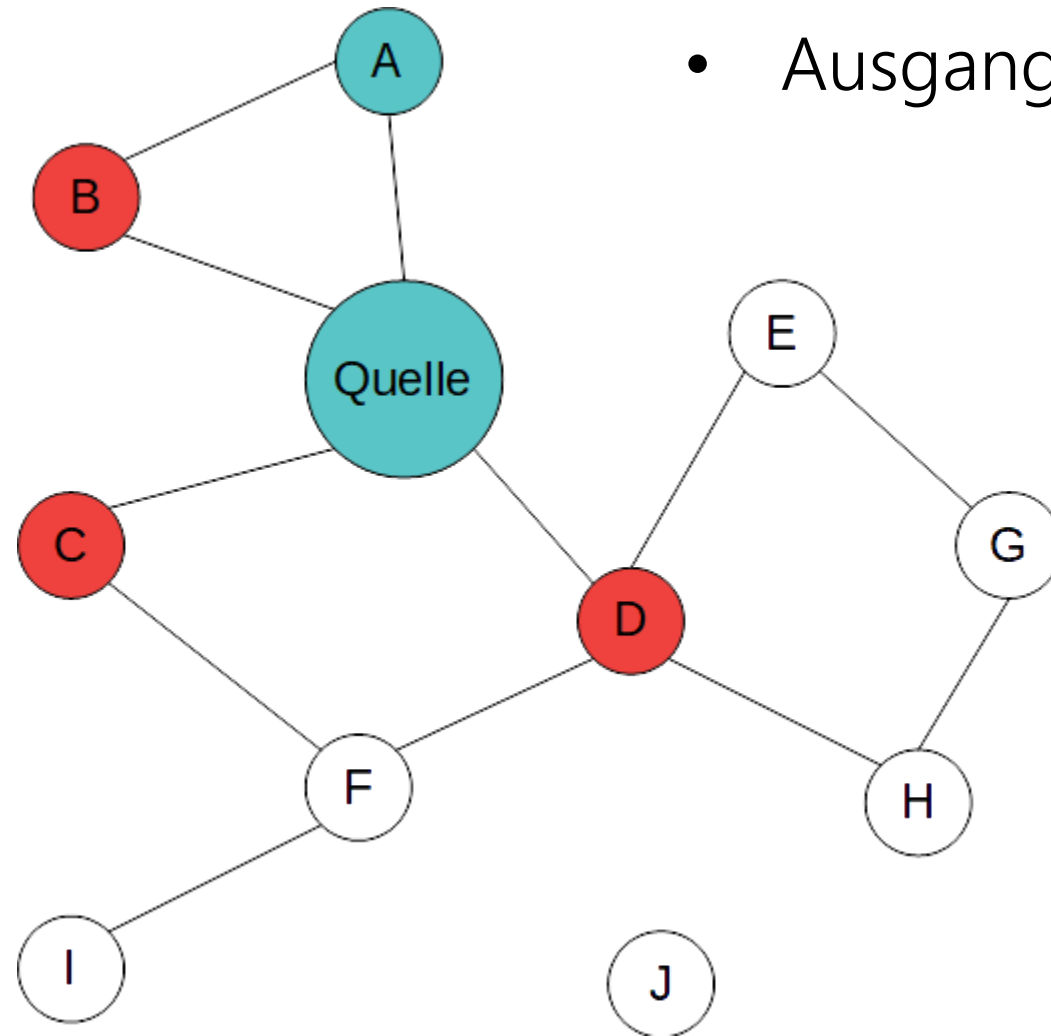


# Breitensuche



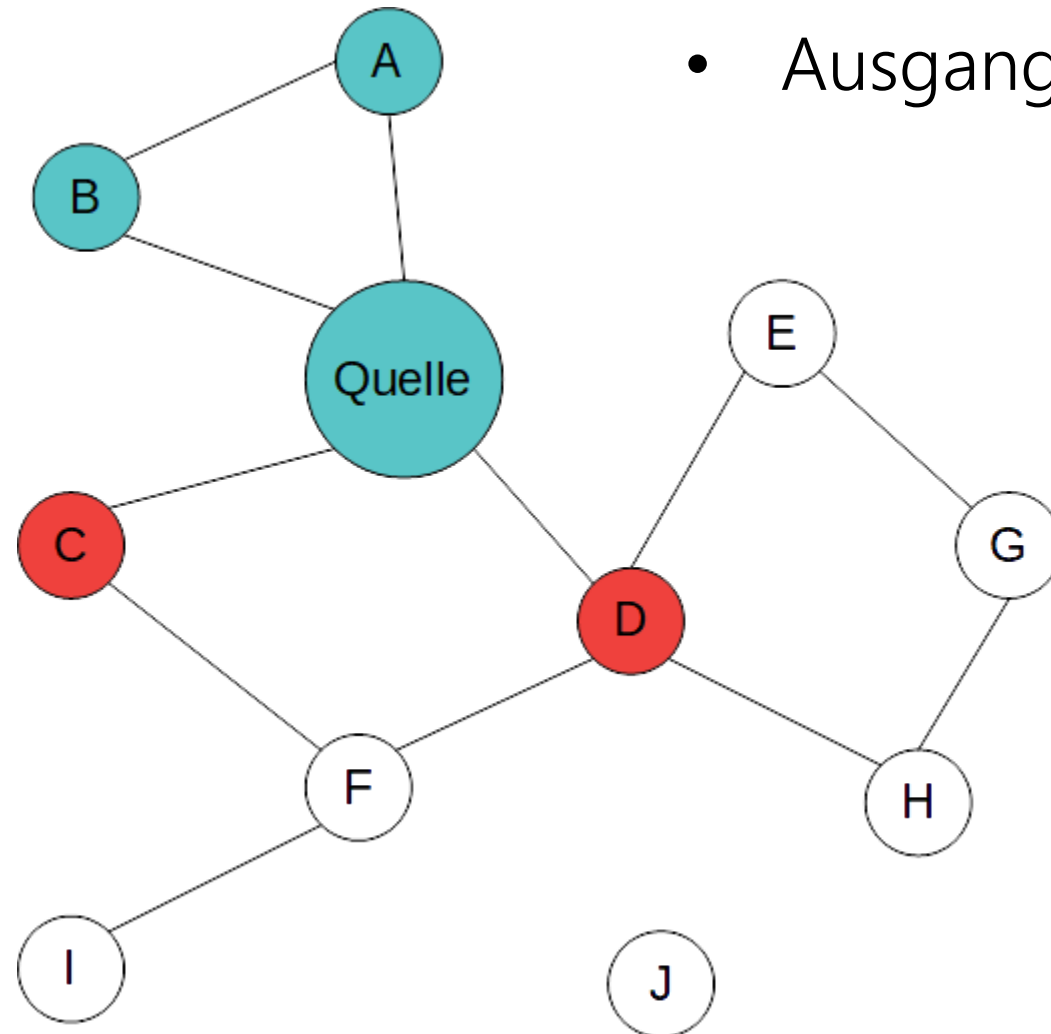
- Ausgangsknoten: A

# Breitensuche



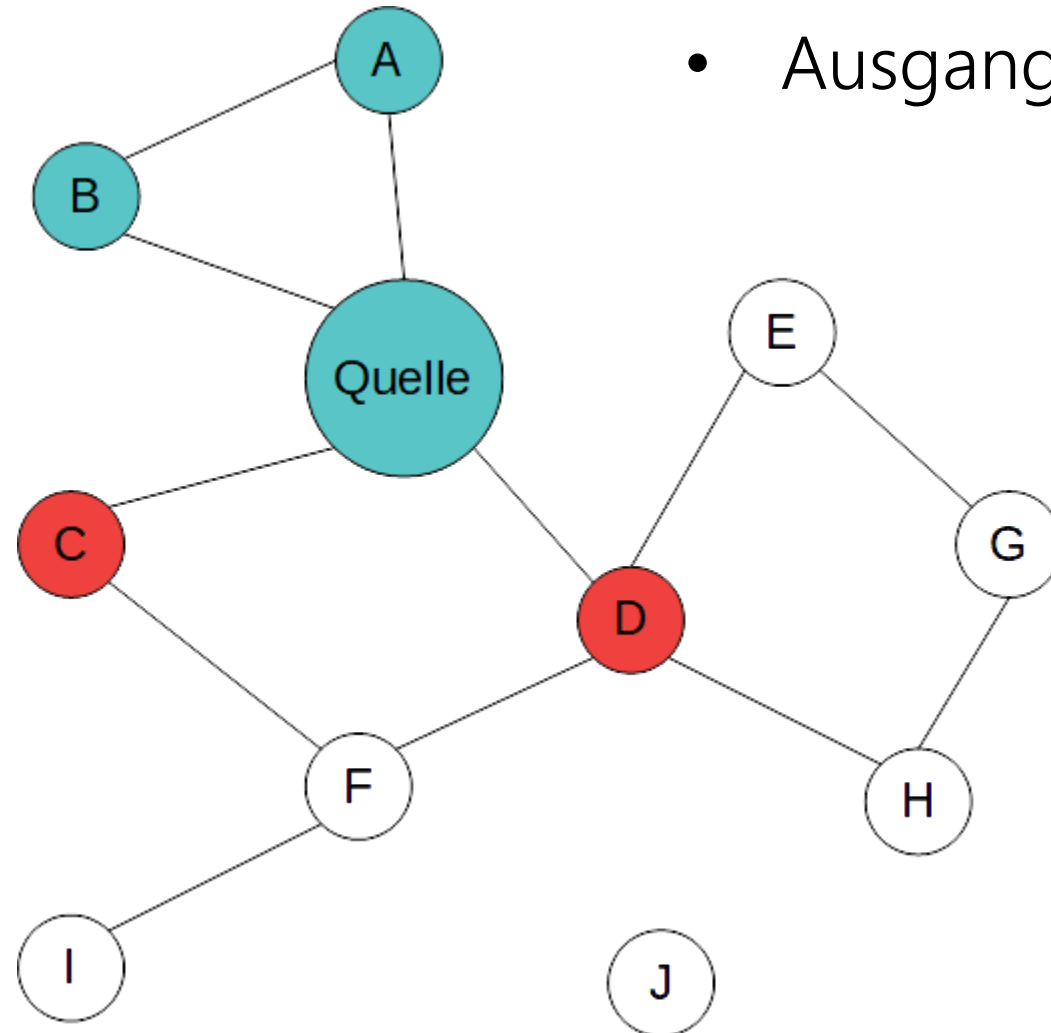
- Ausgangsknoten: B

# Breitensuche



- Ausgangsknoten: B

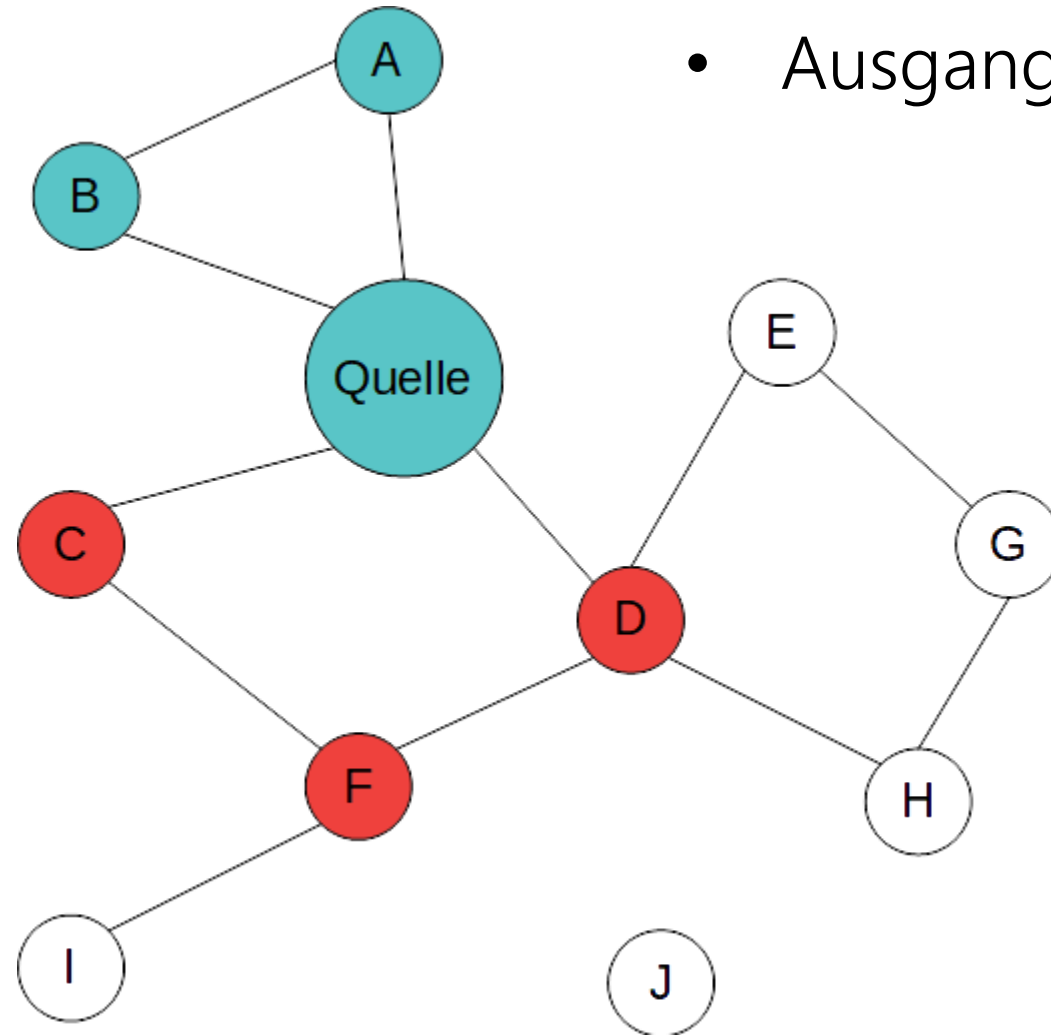
# Breitensuche



- Ausgangsknoten: C

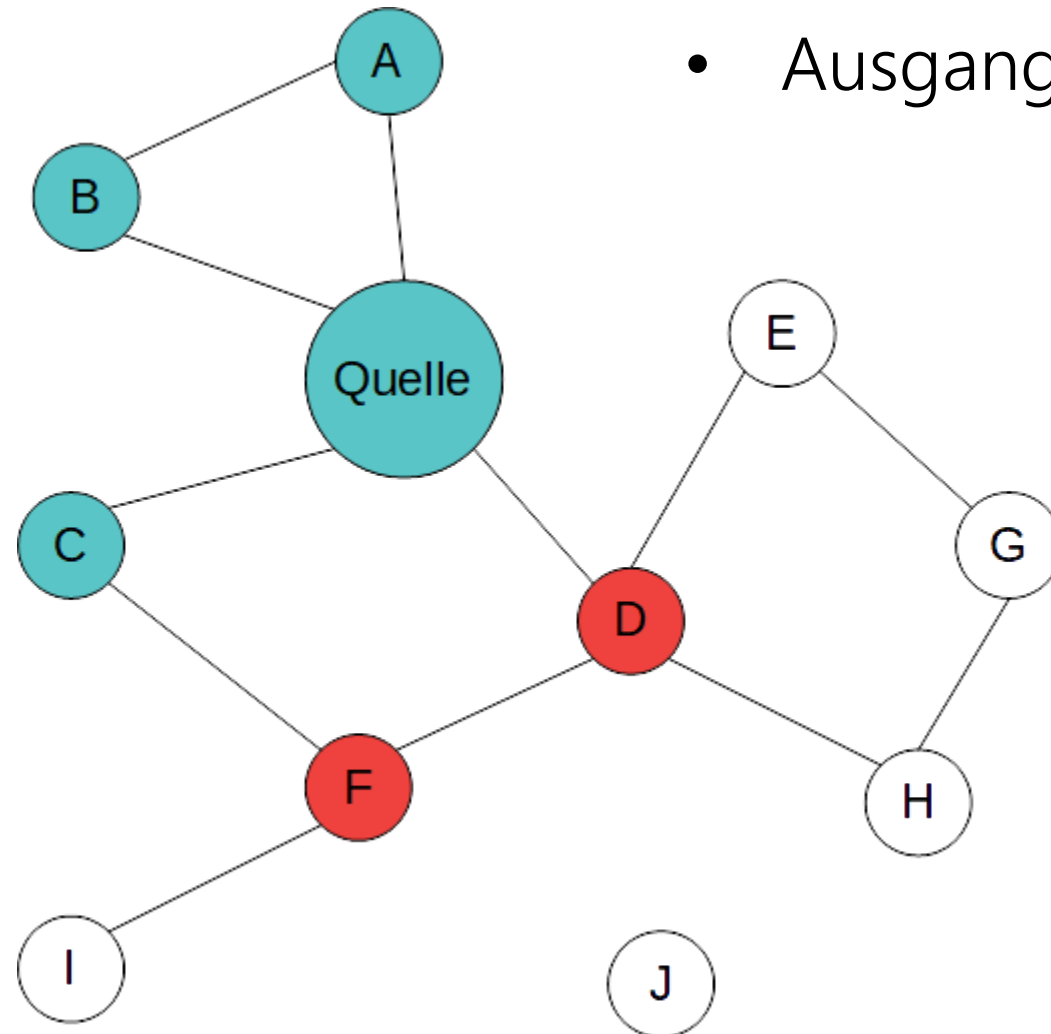


# Breitensuche



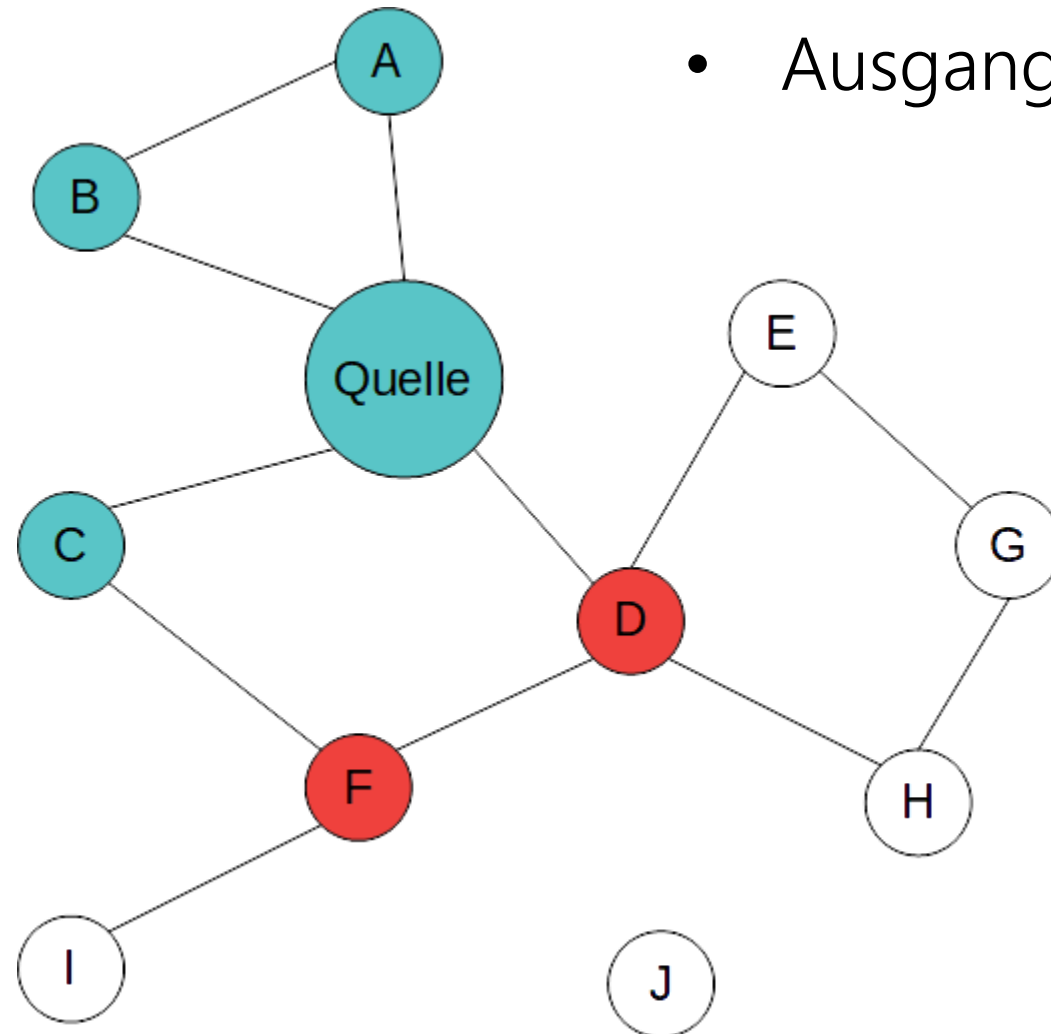
- Ausgangsknoten: C

# Breitensuche



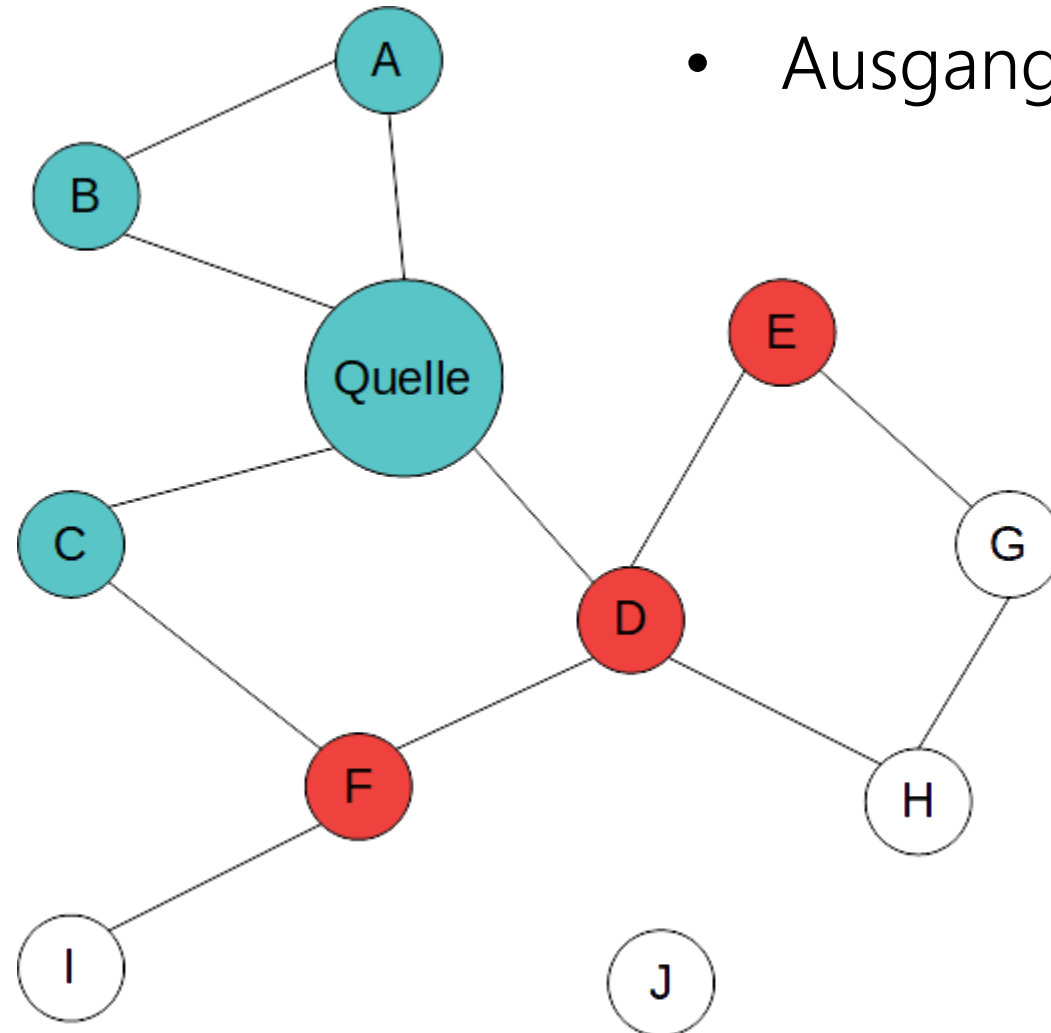
- Ausgangsknoten: C

# Breitensuche



- Ausgangsknoten: D

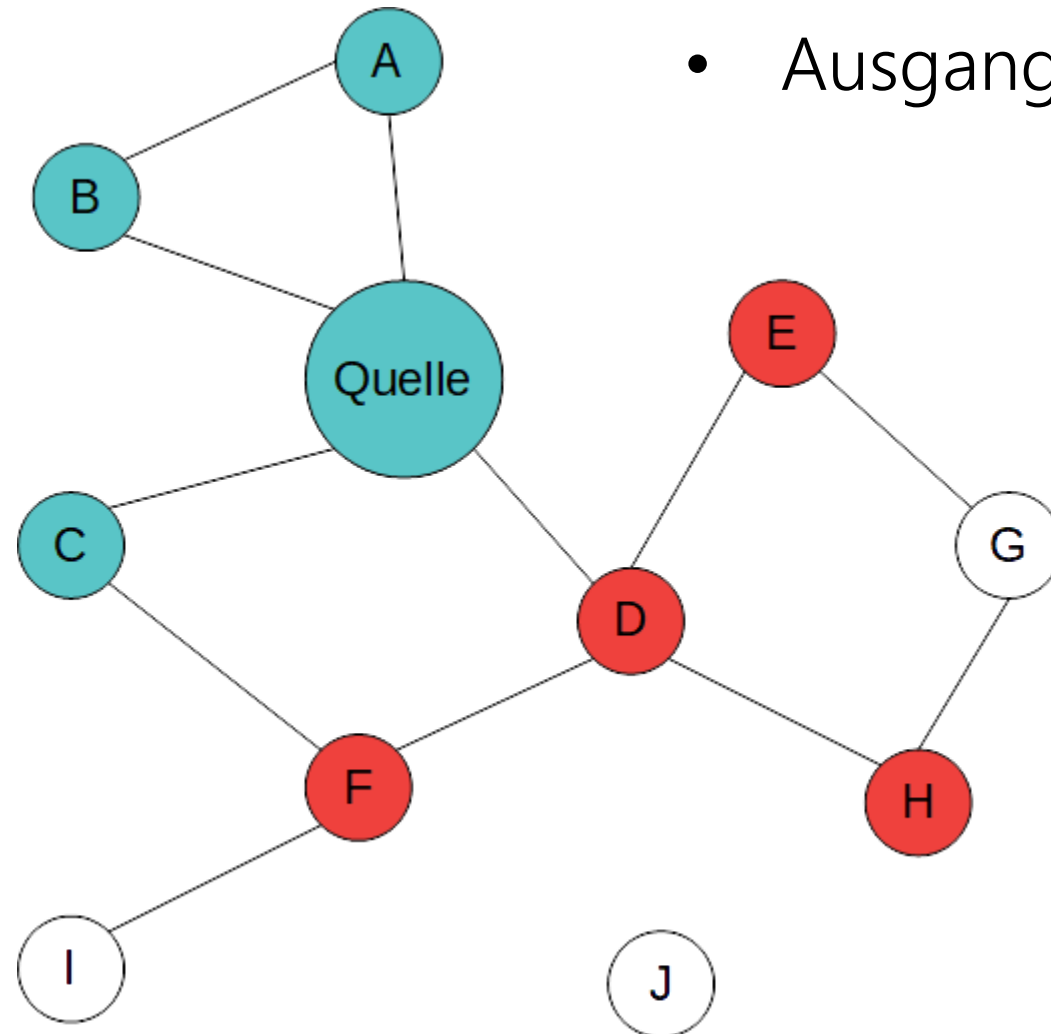
# Breitensuche



- Ausgangsknoten: D

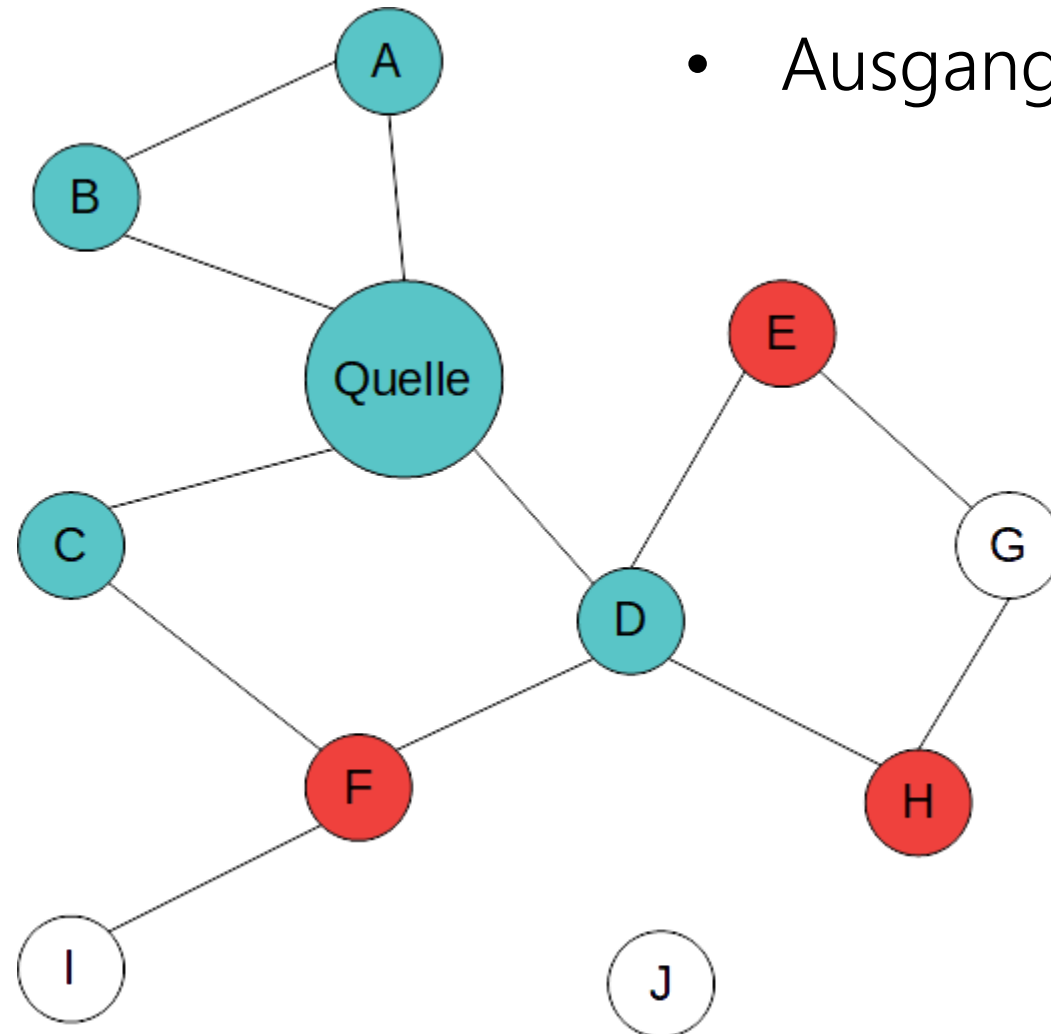


# Breitensuche



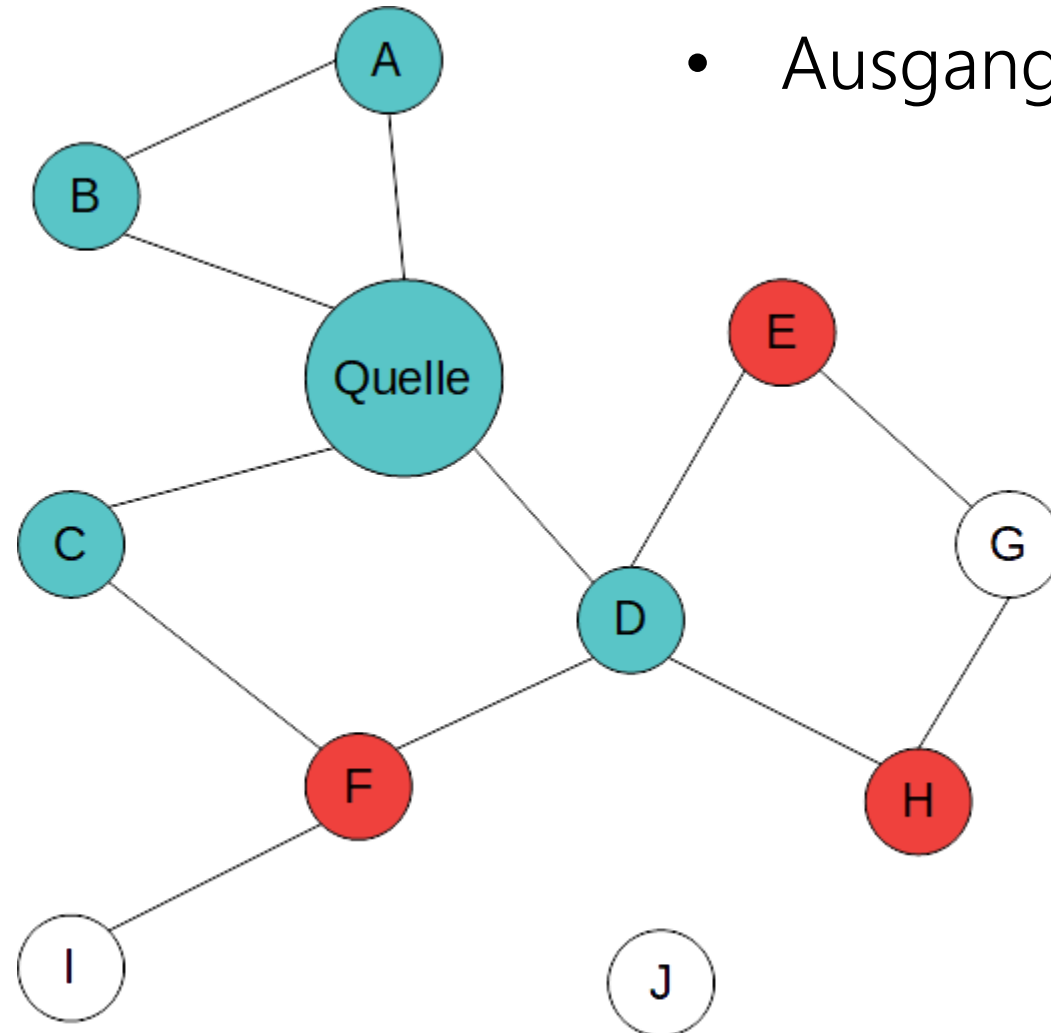
- Ausgangsknoten: D

# Breitensuche



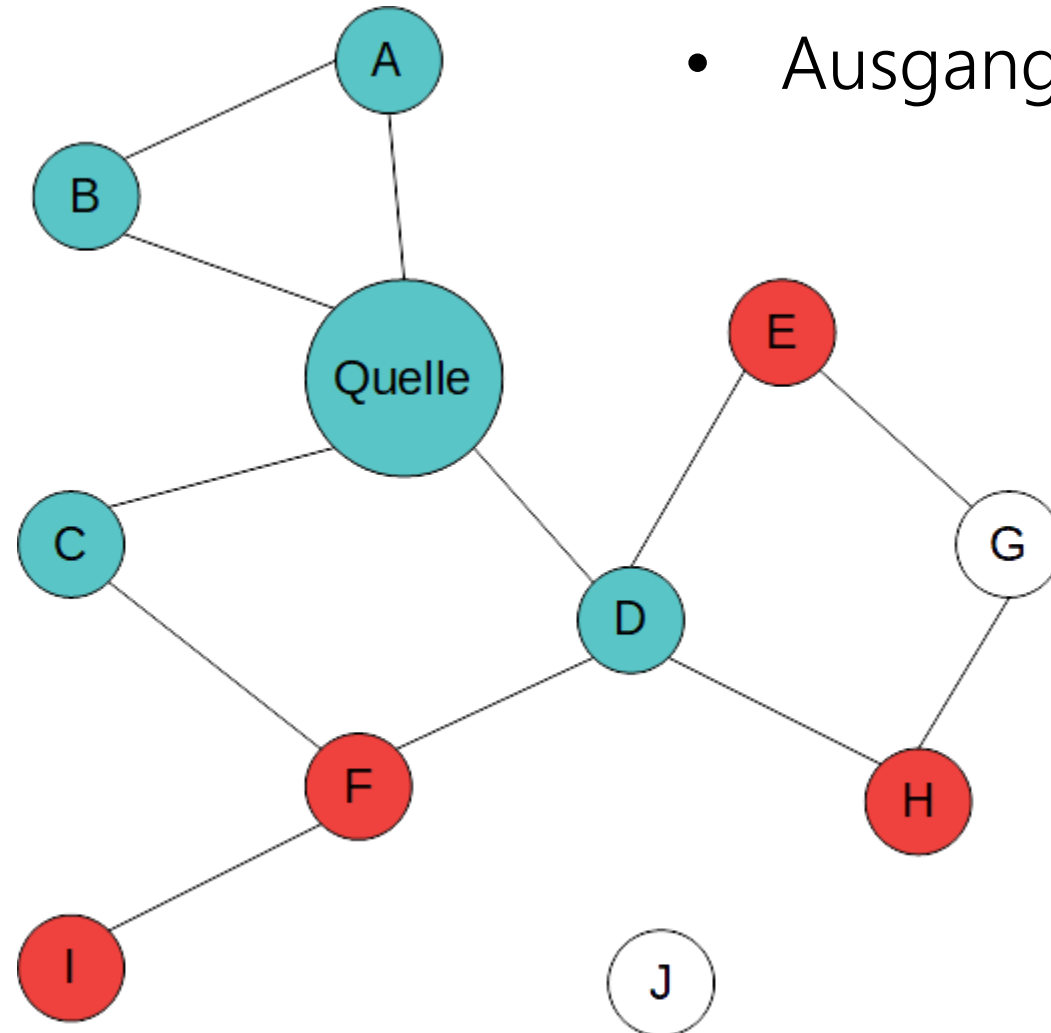
- Ausgangsknoten: D

# Breitensuche



- Ausgangsknoten: F

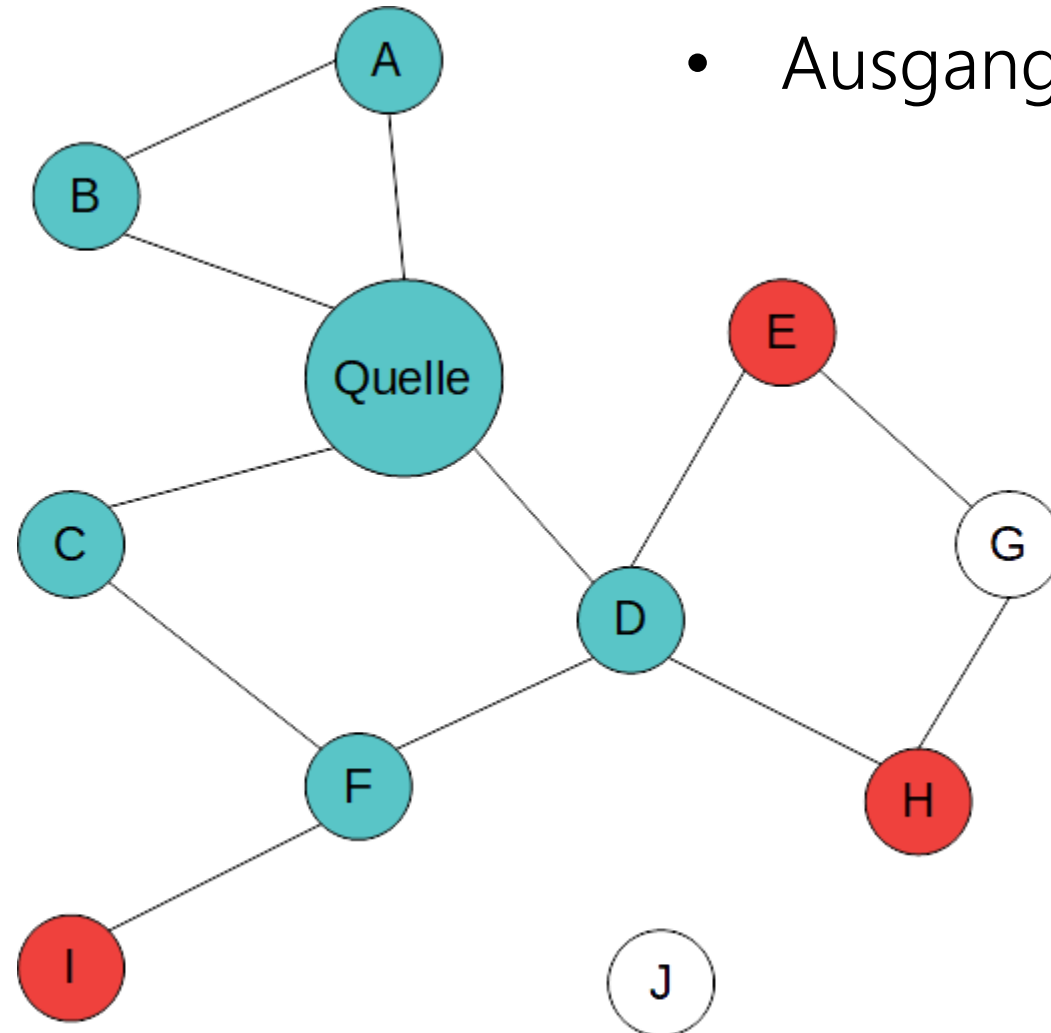
# Breitensuche



- Ausgangsknoten: F

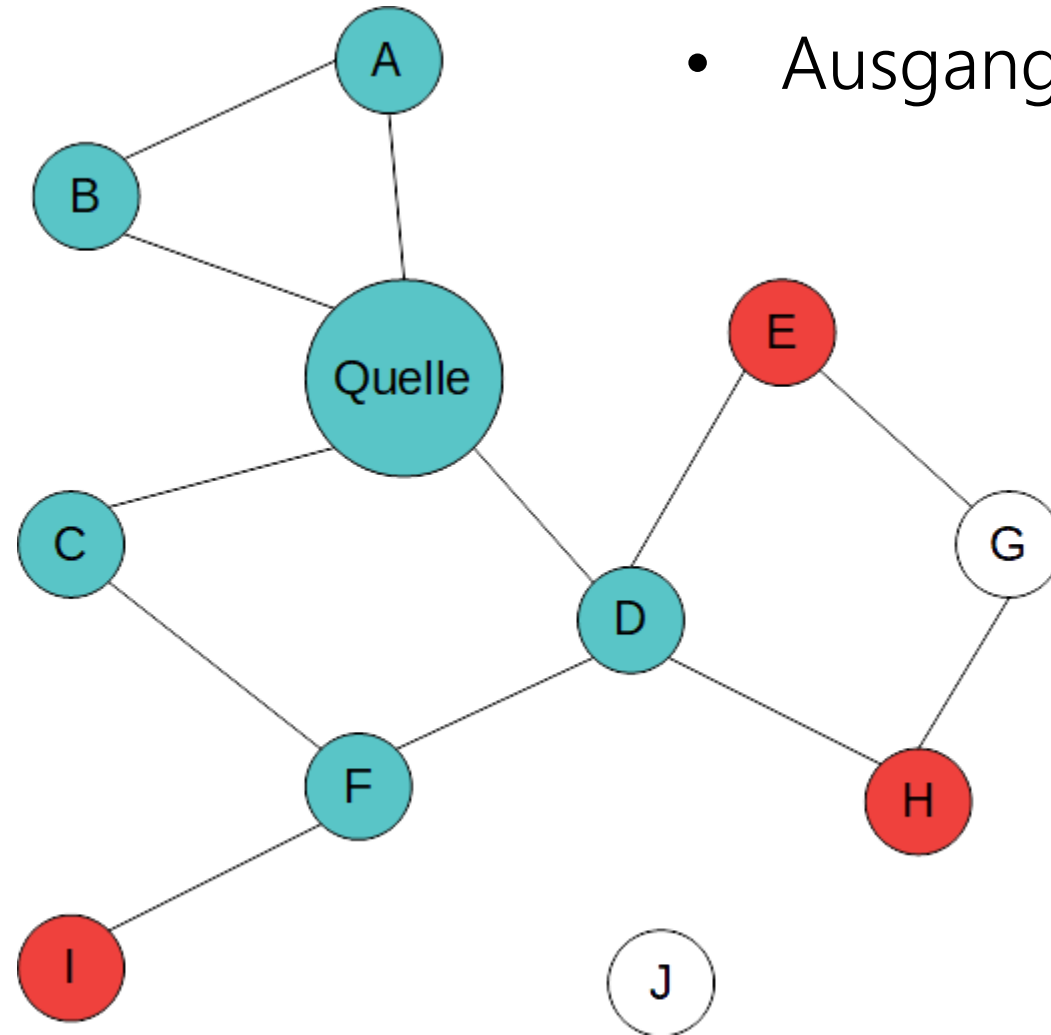


# Breitensuche



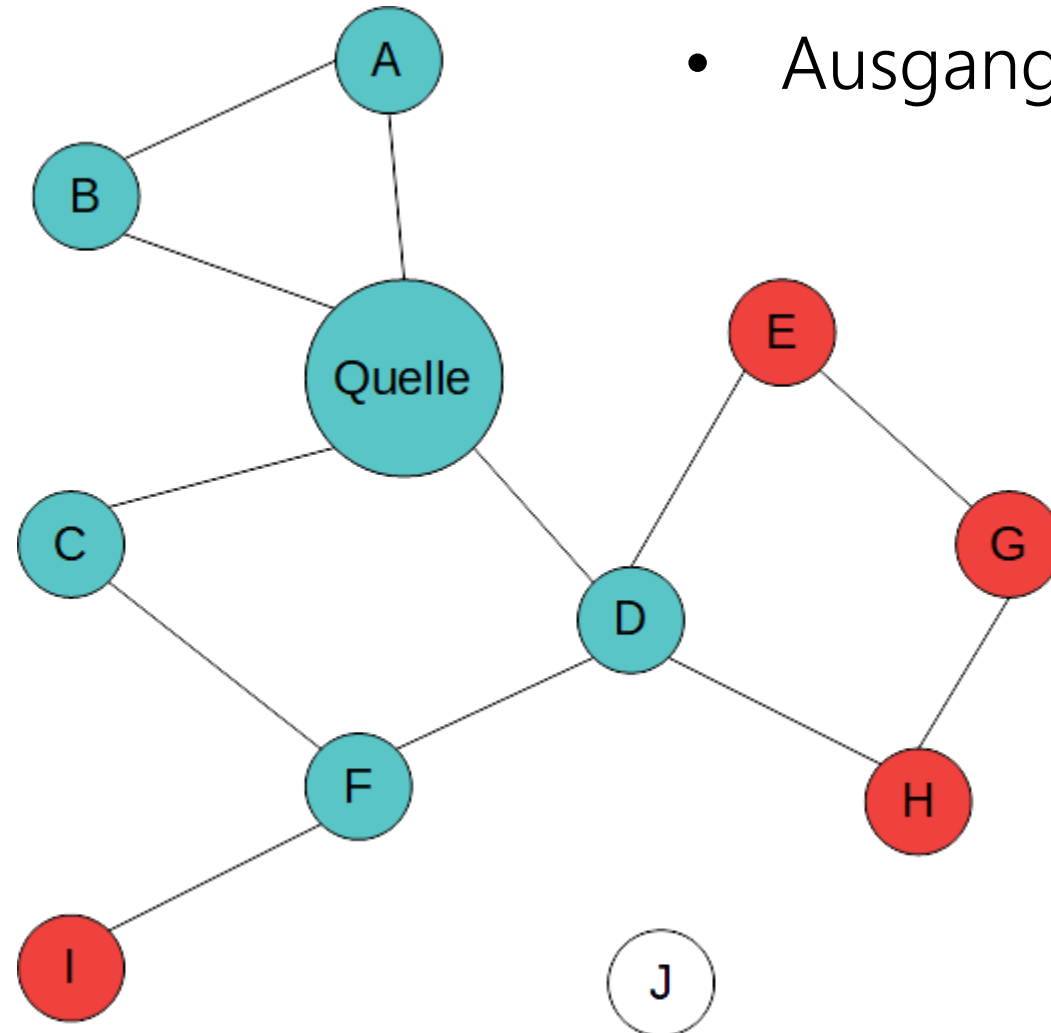
- Ausgangsknoten: F

# Breitensuche



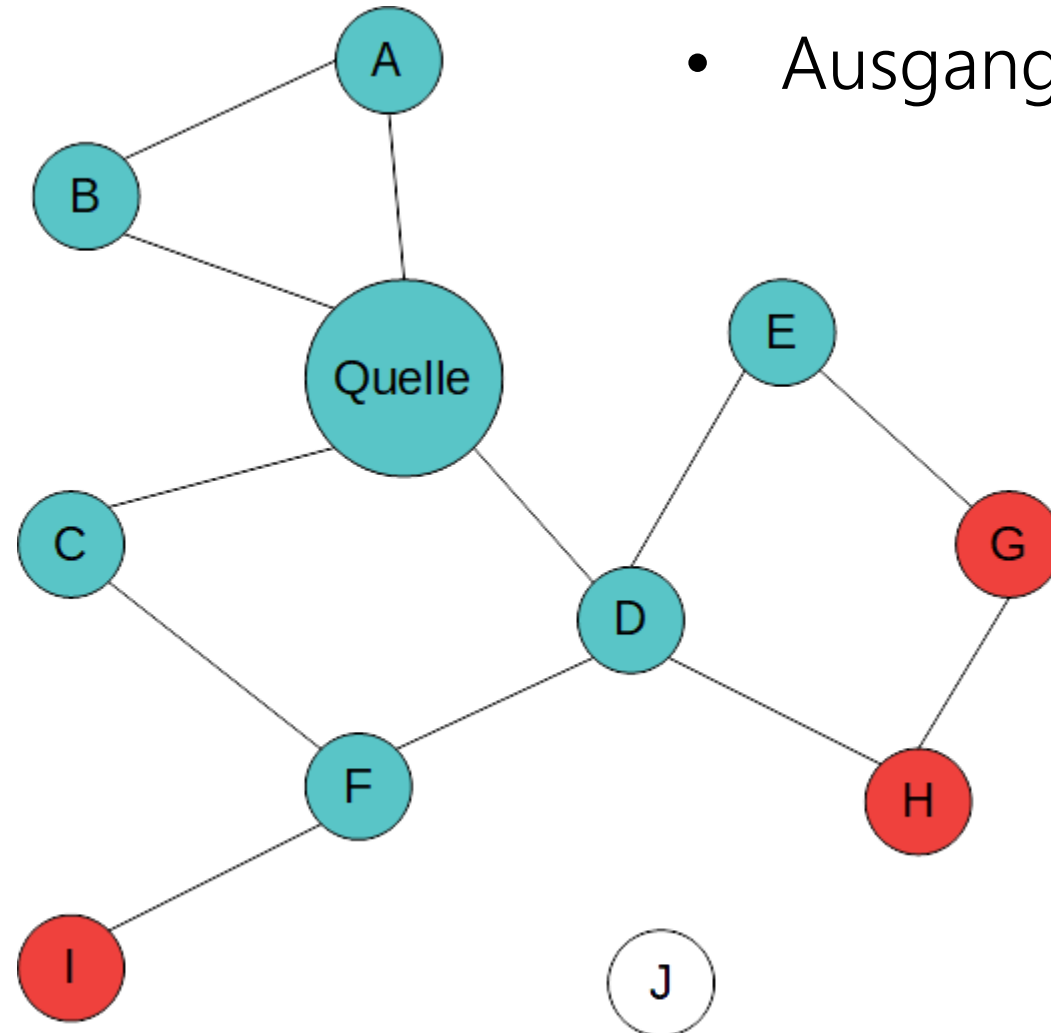
- Ausgangsknoten: E

# Breitensuche



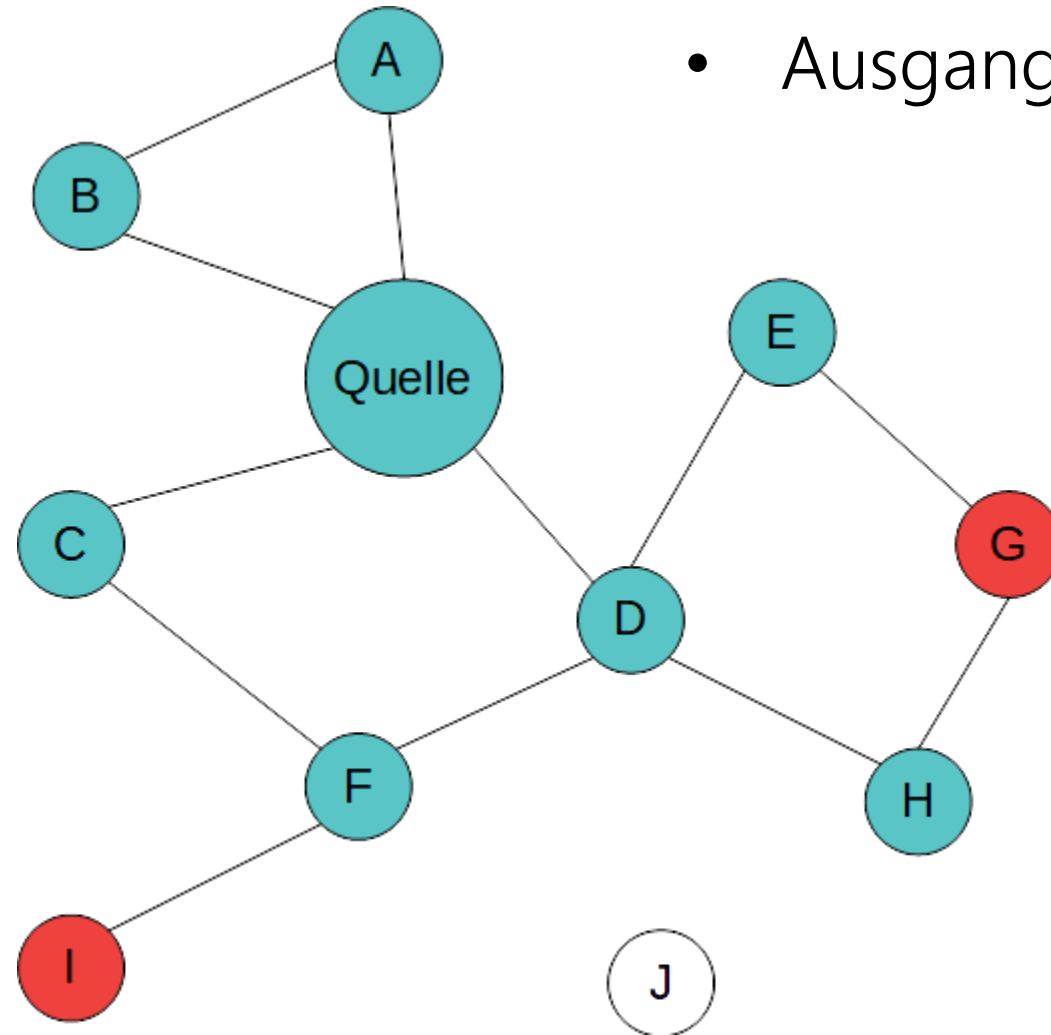
- Ausgangsknoten: E

# Breitensuche



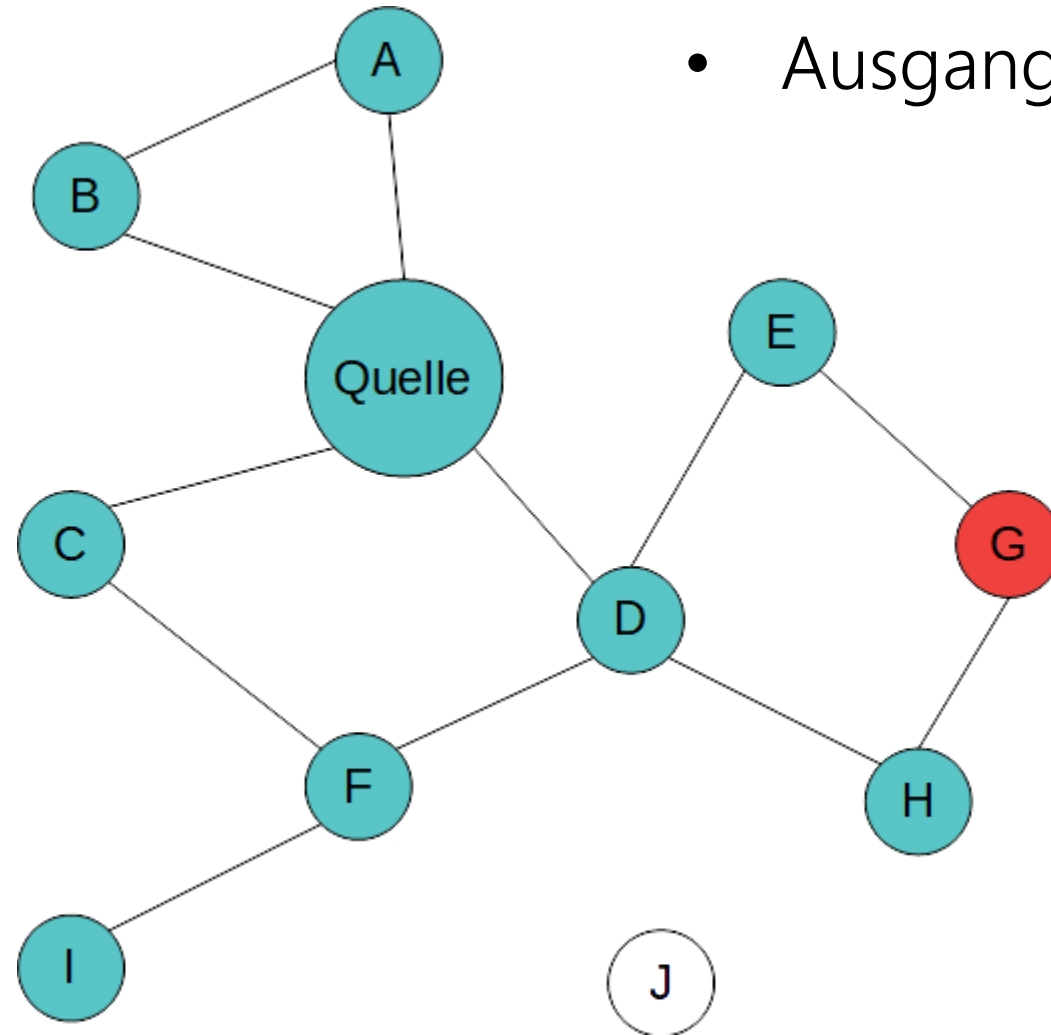
- Ausgangsknoten: E

# Breitensuche



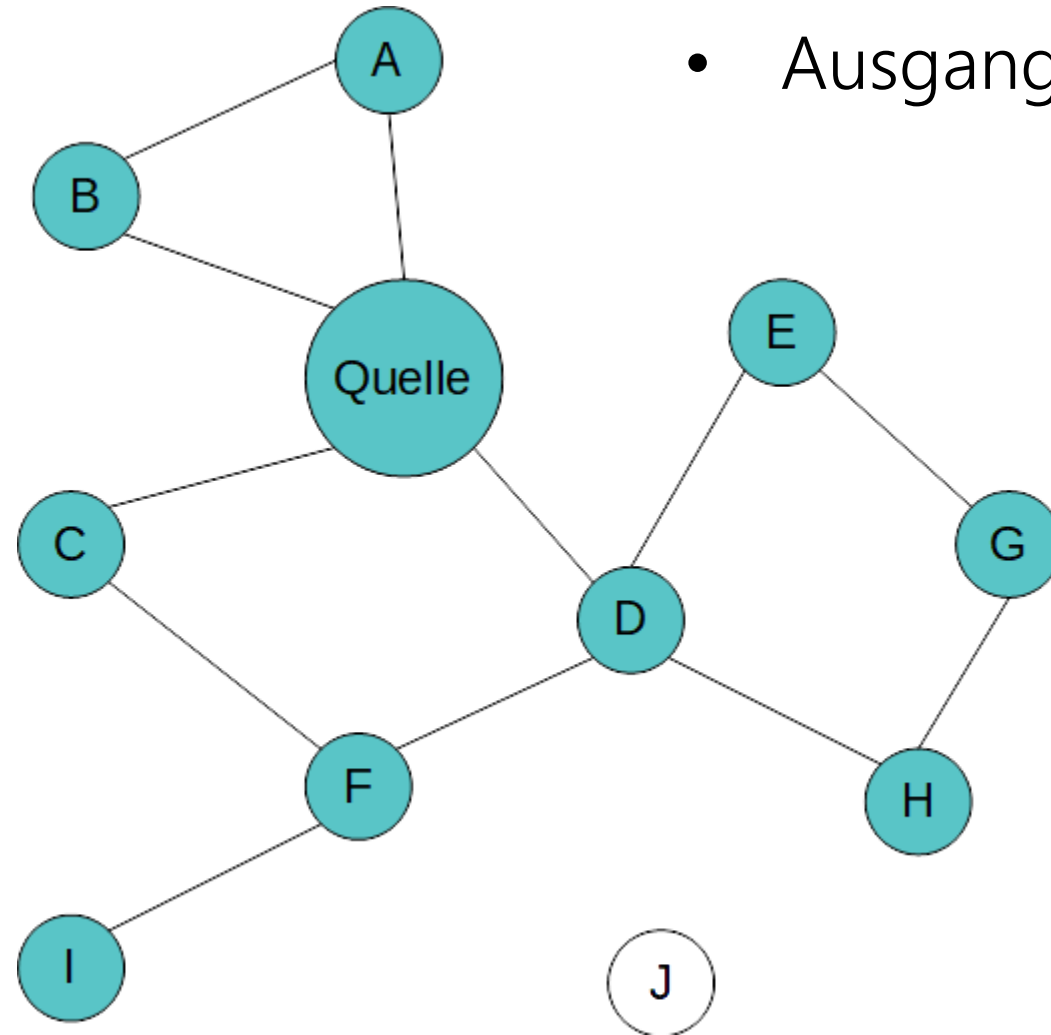
- Ausgangsknoten: H

# Breitensuche



- Ausgangsknoten: I

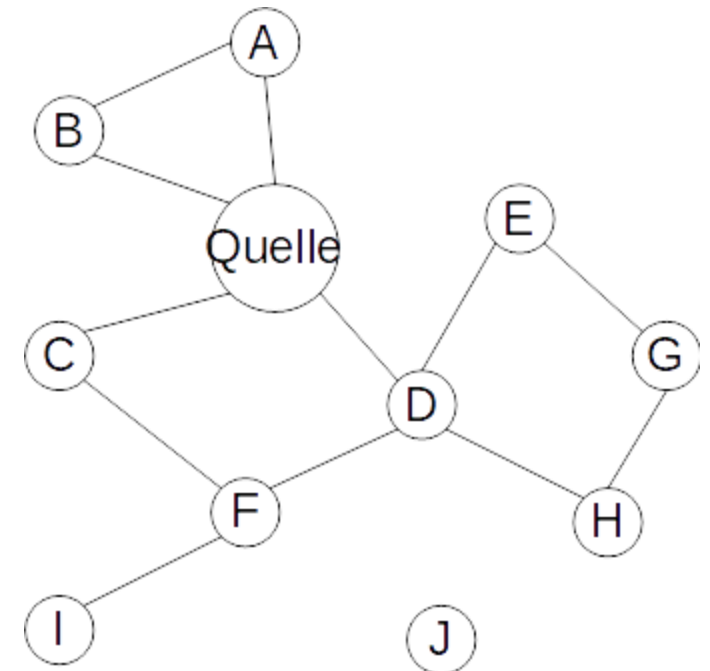
# Breitensuche



- Ausgangsknoten: G

# Strategie der Breitensuche

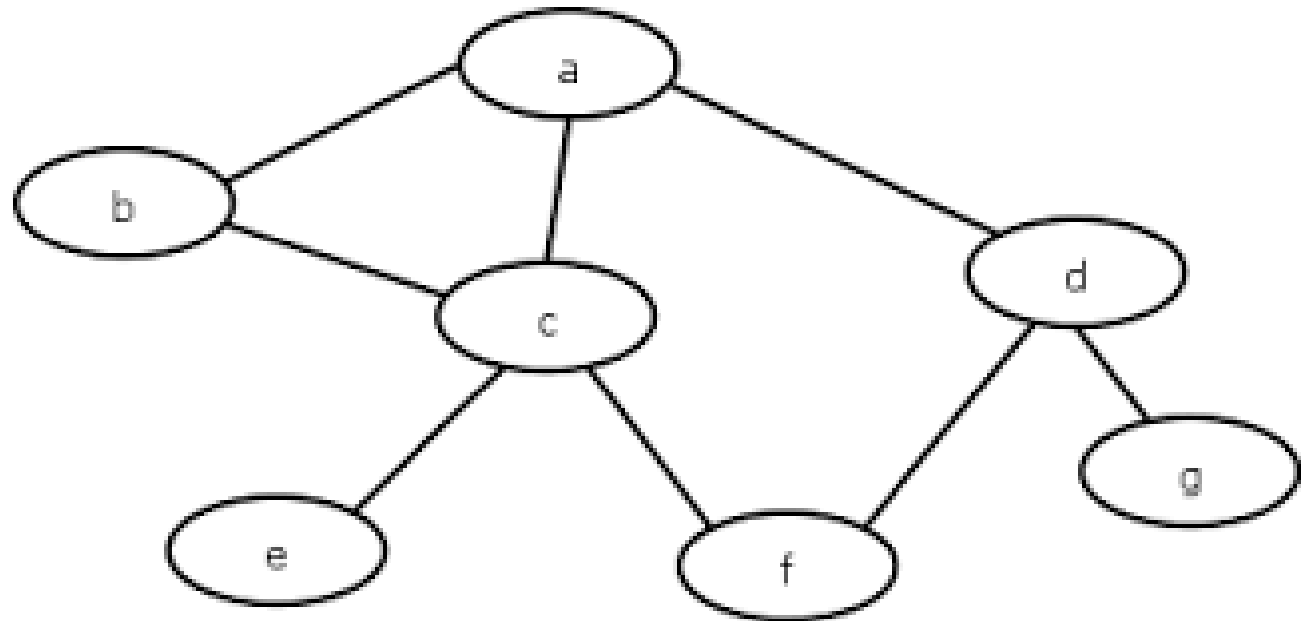
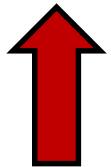
- Jeder Knoten kann auf **besucht** und **fertig** gesetzt werden.
- Besuche vom Ausgangsknoten jeden unbesuchten Knoten
- Sind von diesem Knoten alle Nachbarn besucht, so wird der nächste Ausgangsknoten der als erstes besuchte.





# Übung Breitensuche

- Führe den Breitendurchlauf zu folgendem Graphen durch:
  - Startknoten ist b.
  - Gib alle nacheinander besuchten Knoten an.
  - Befülle das Merke-Array.



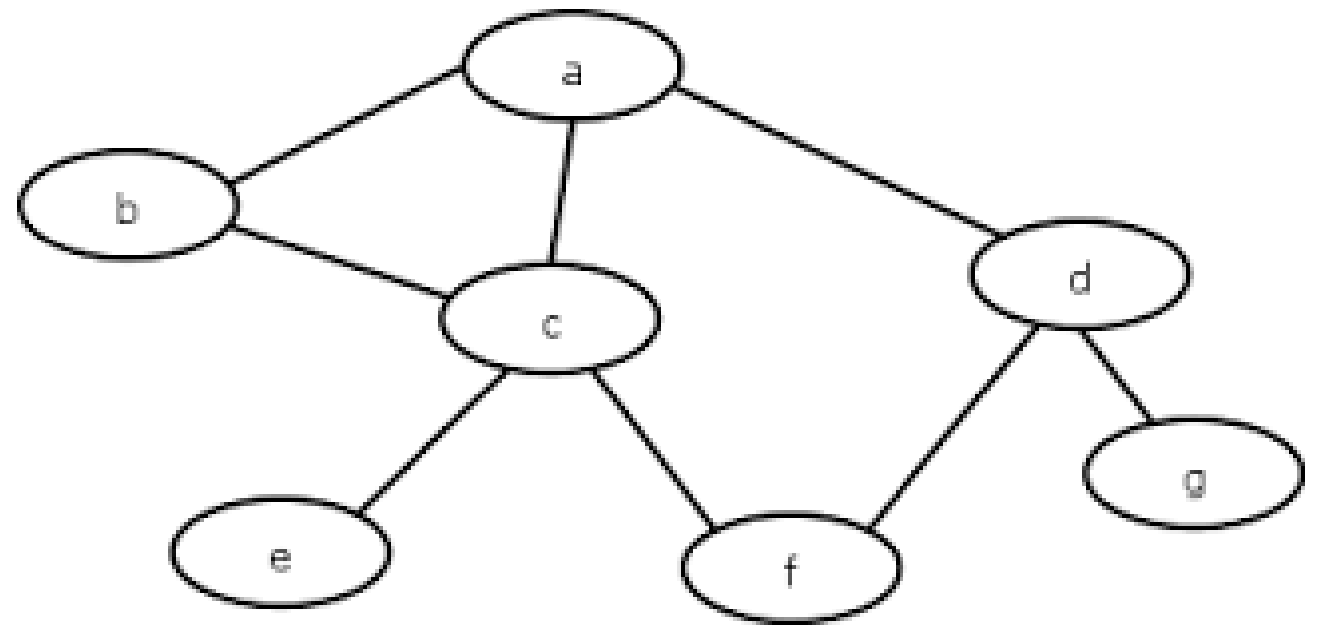
# Übung Breitensuche

- Der Index gibt die nächste freie Position im Merke-Array an



• index = 1

- Besucht:  
b

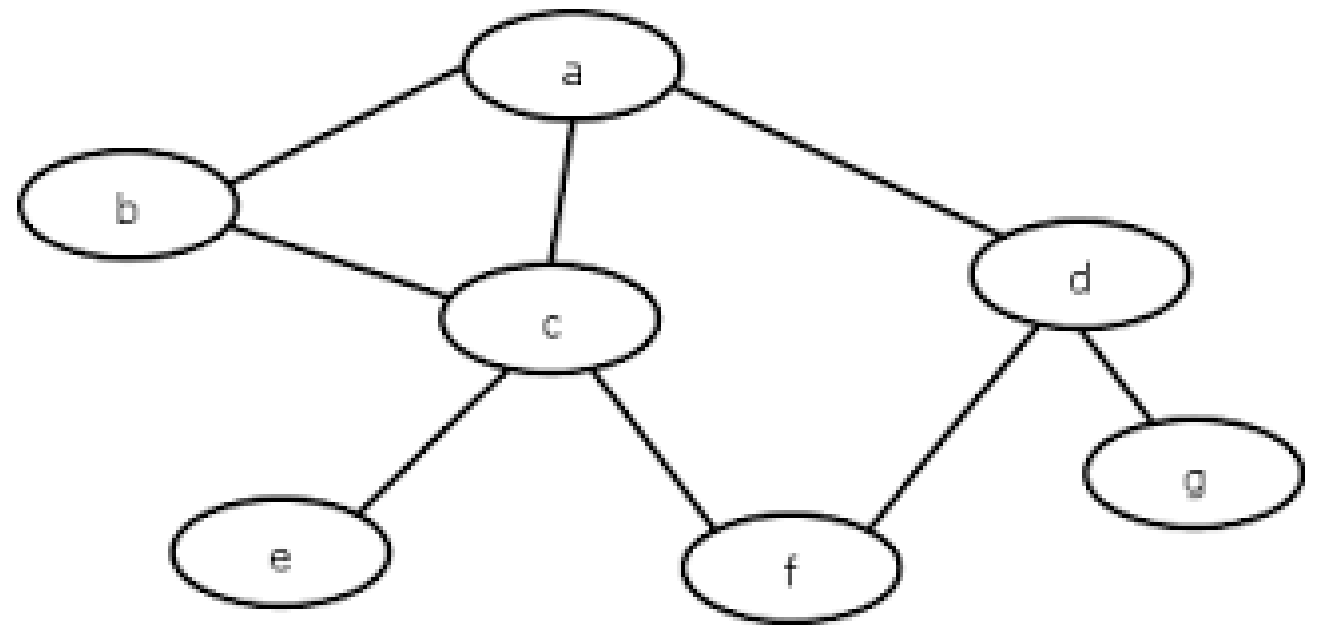


# Übung Breitensuche



• index = 2

- Besucht:  
b – a – ...



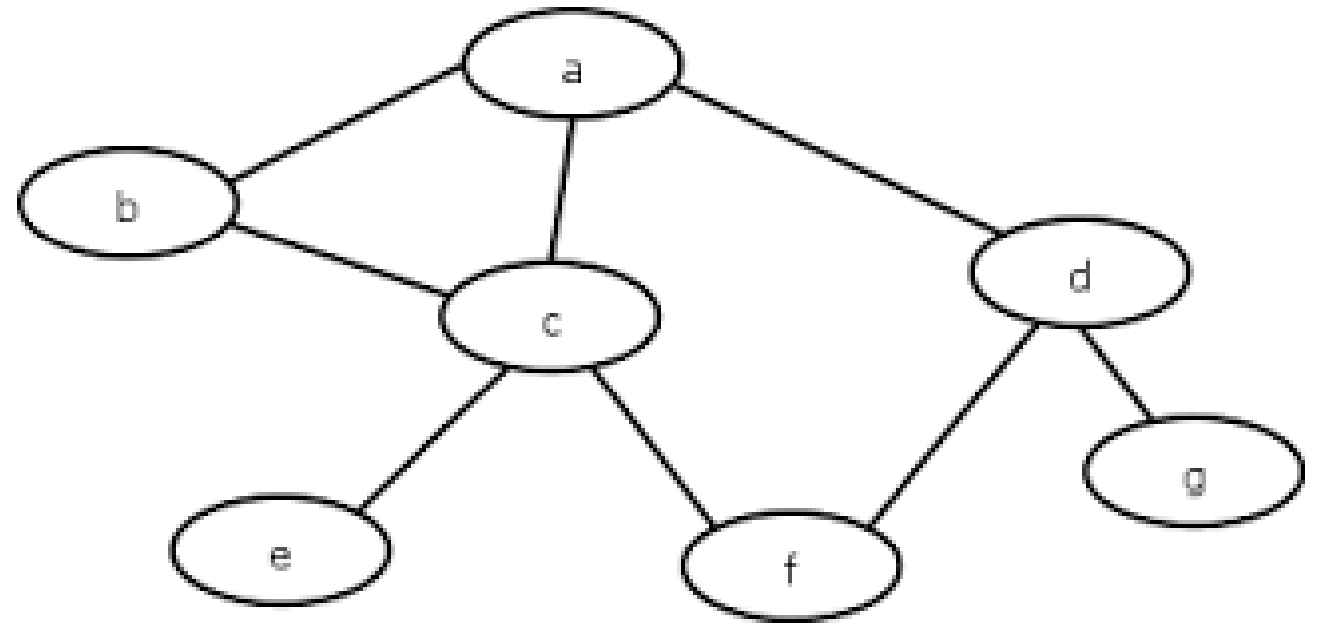
# Übung Breitensuche



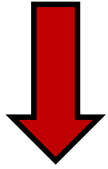
• index = 3

• Besucht:

b – a – c

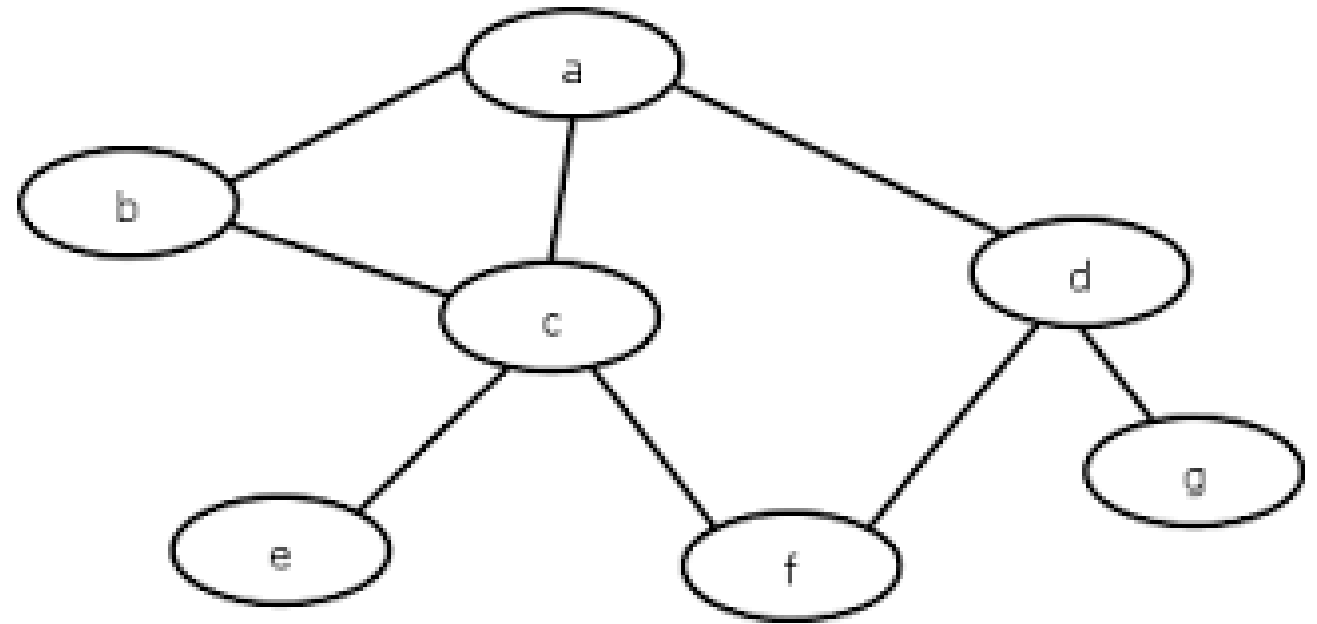


# Übung Breitensuche



- index = 3

- Besucht:  
b – a – c – ...

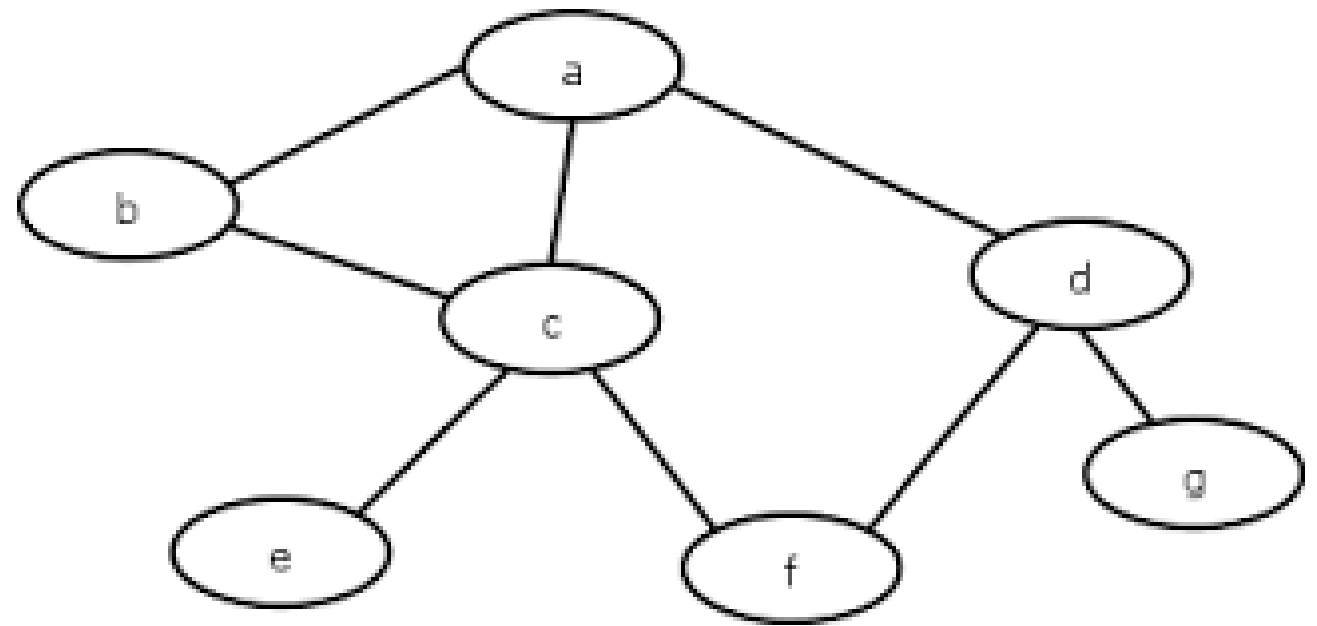


# Übung Breitensuche



- index = 4

- Besucht:  
b – a – c – d

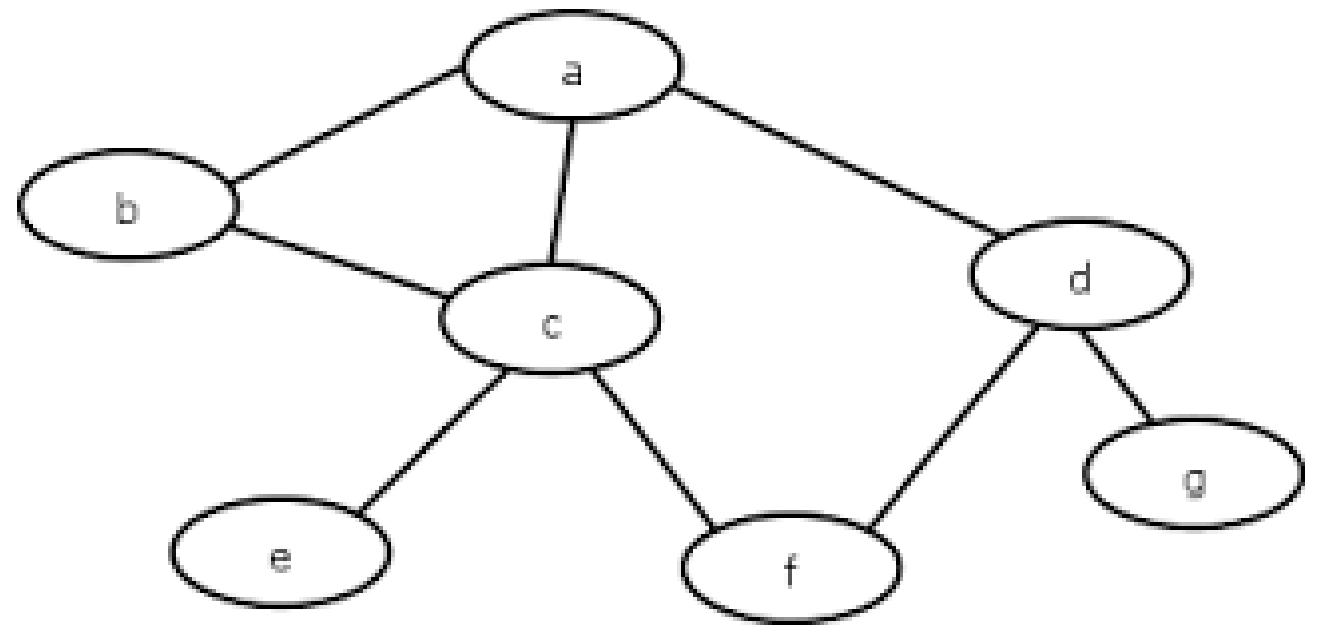


# Übung Breitensuche

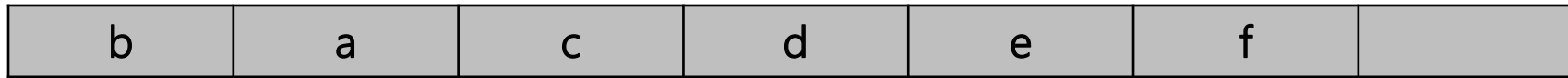


- index = 4

- Besucht:  
b – a – c – d



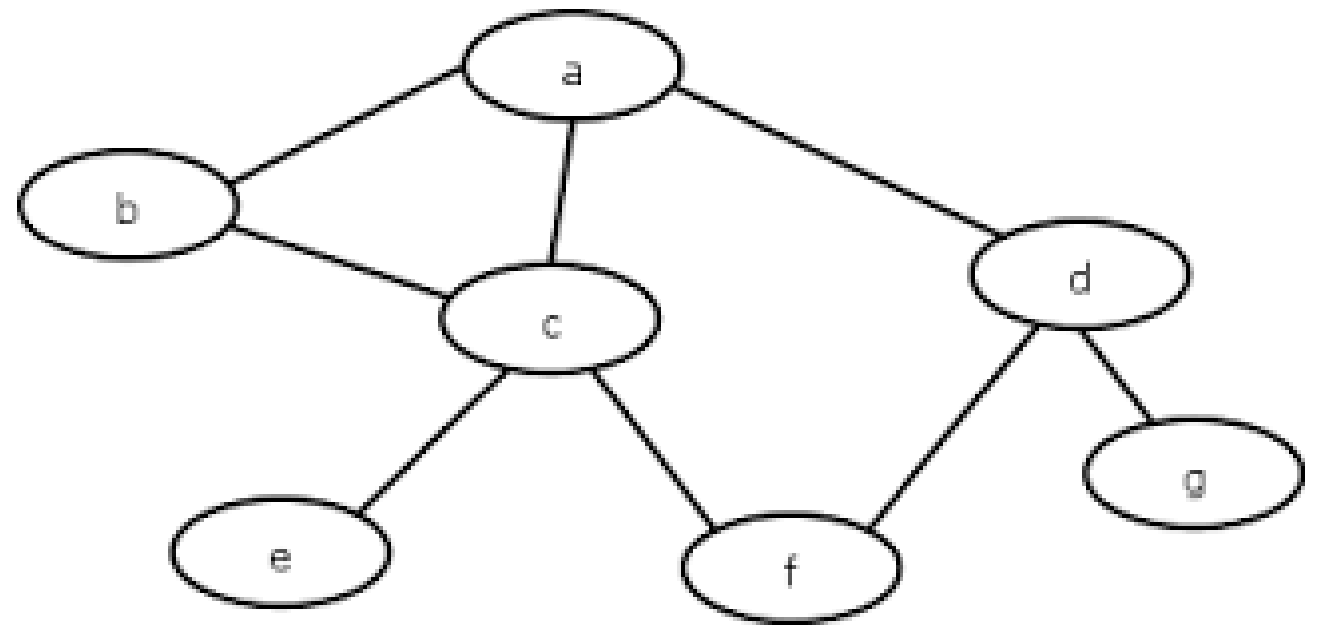
# Übung Breitensuche



• index = 6

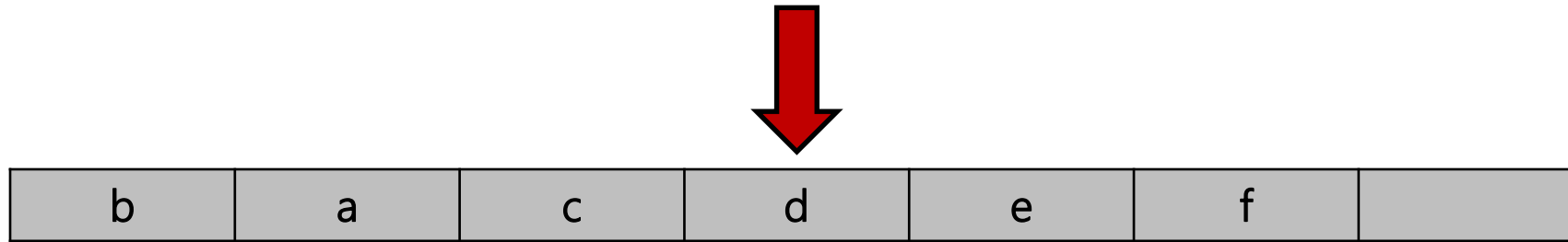
• Besucht:

b – a – c – d – e – f





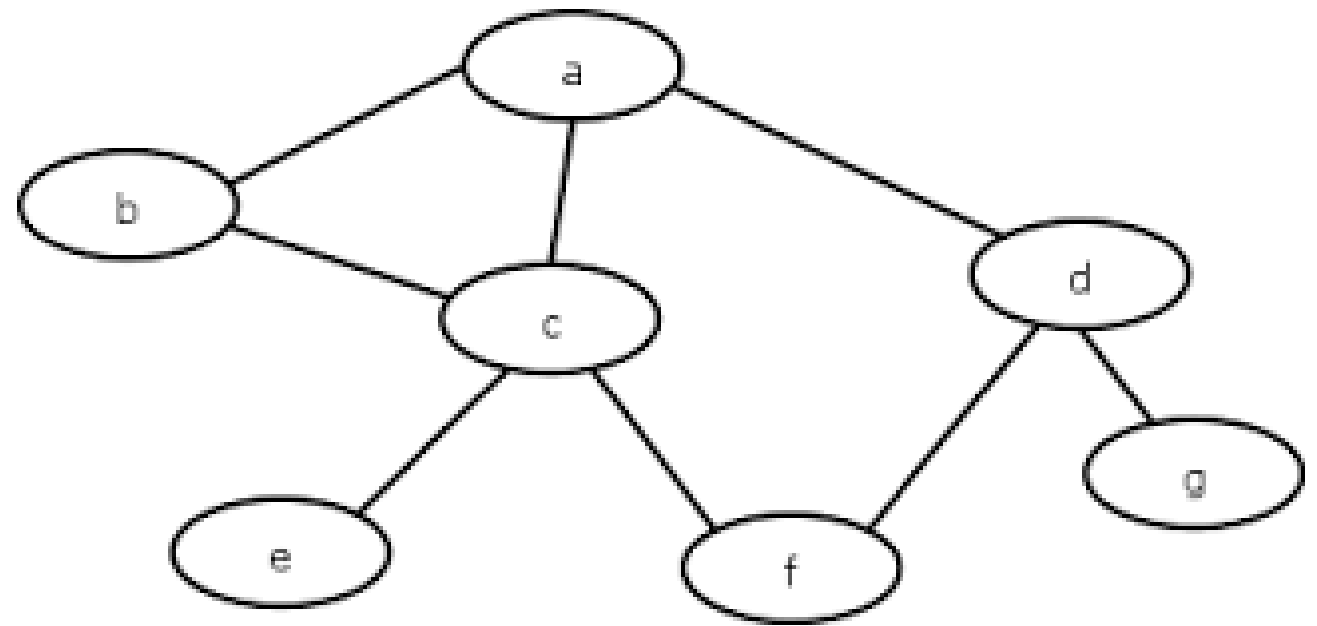
# Übung Breitensuche



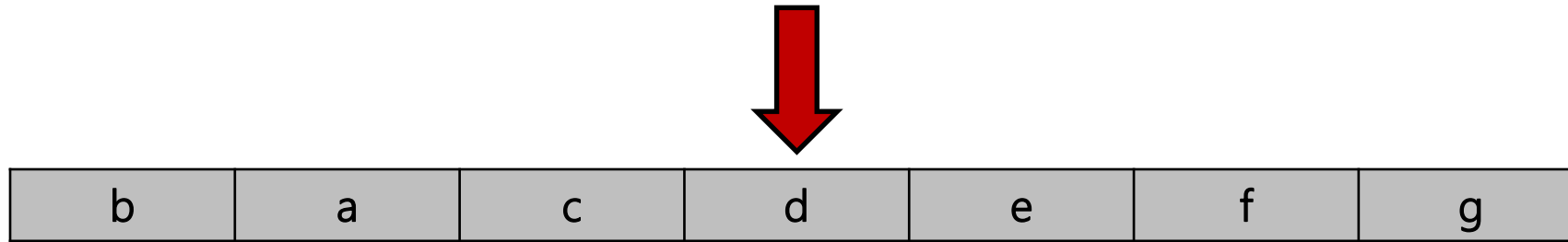
- index = 6

- Besucht:

b – a – c – d – e – f



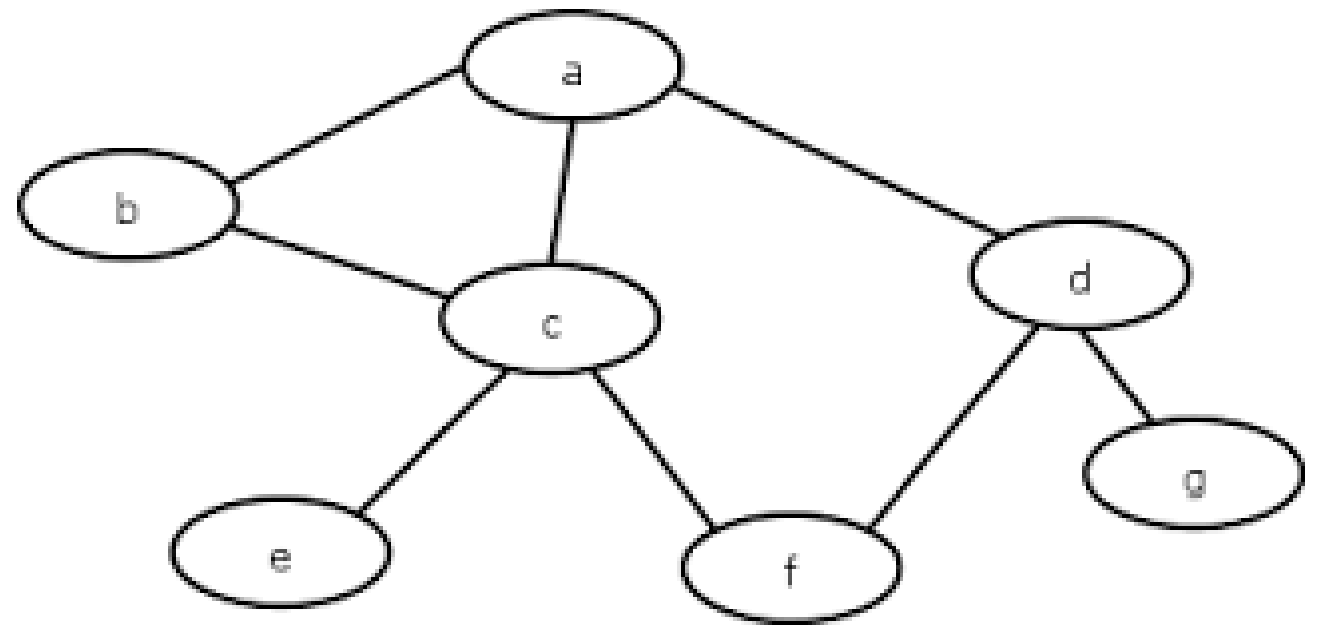
# Übung Breitensuche



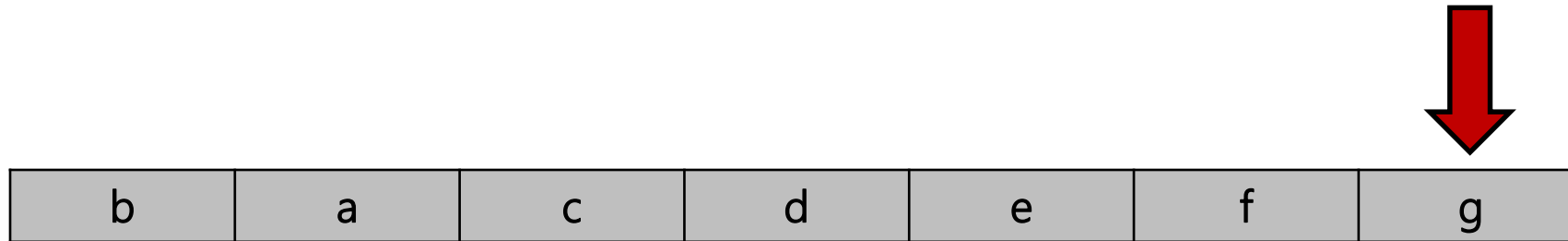
- index = 7

- Besucht:

b – a – c – d – e – f – g



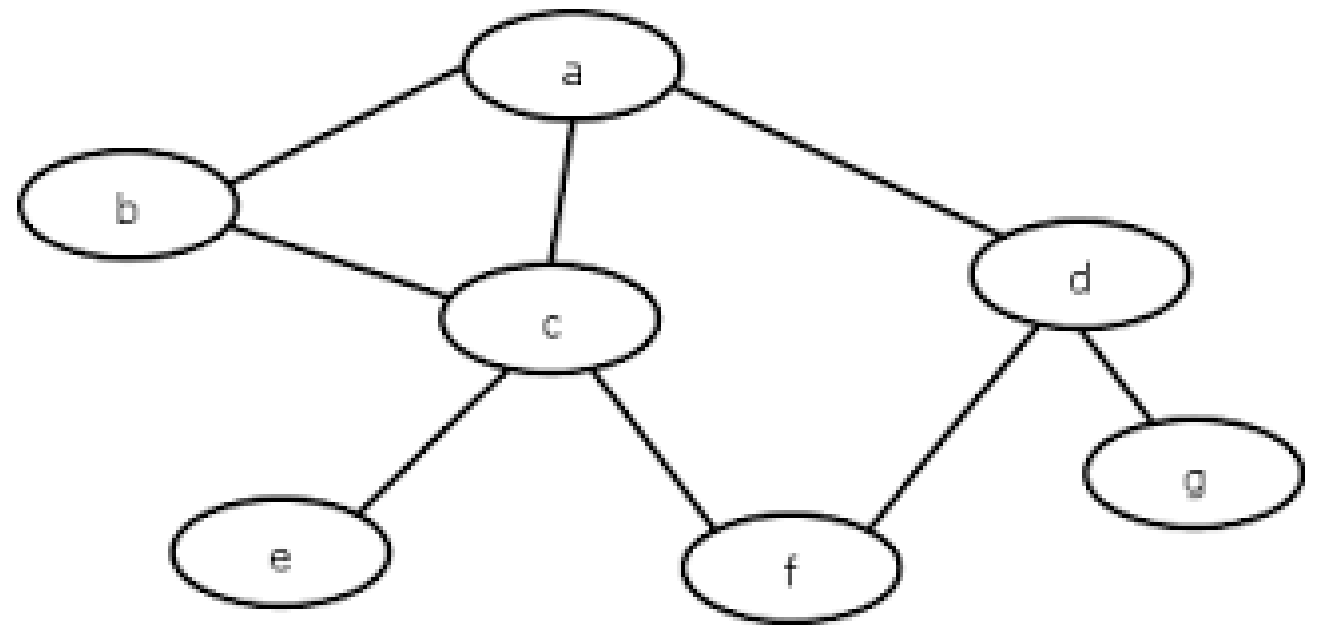
# Übung Breitensuche



- index = 7

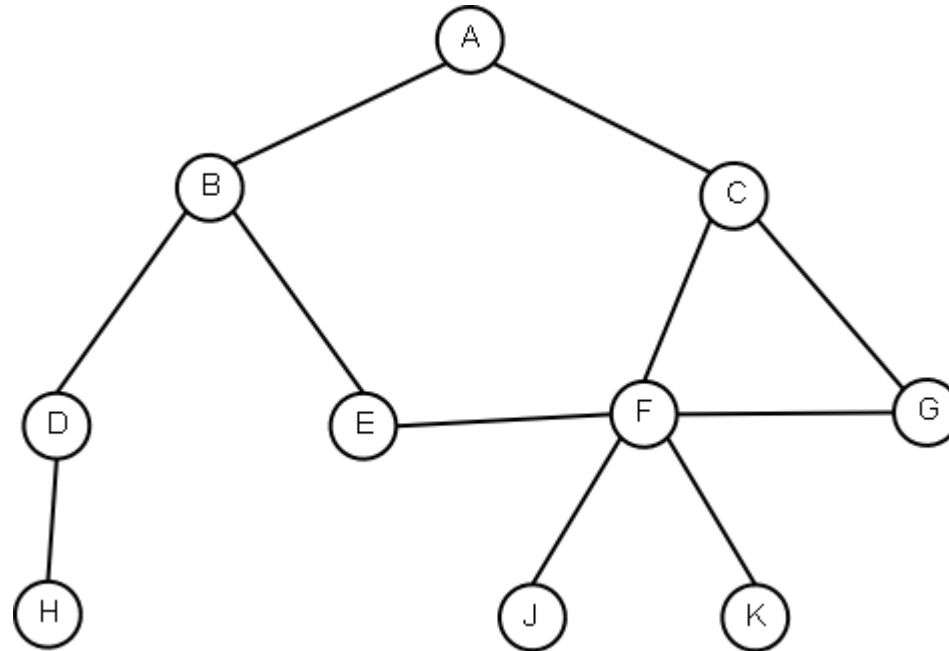
- Besucht:

b – a – c – d – e – f – g



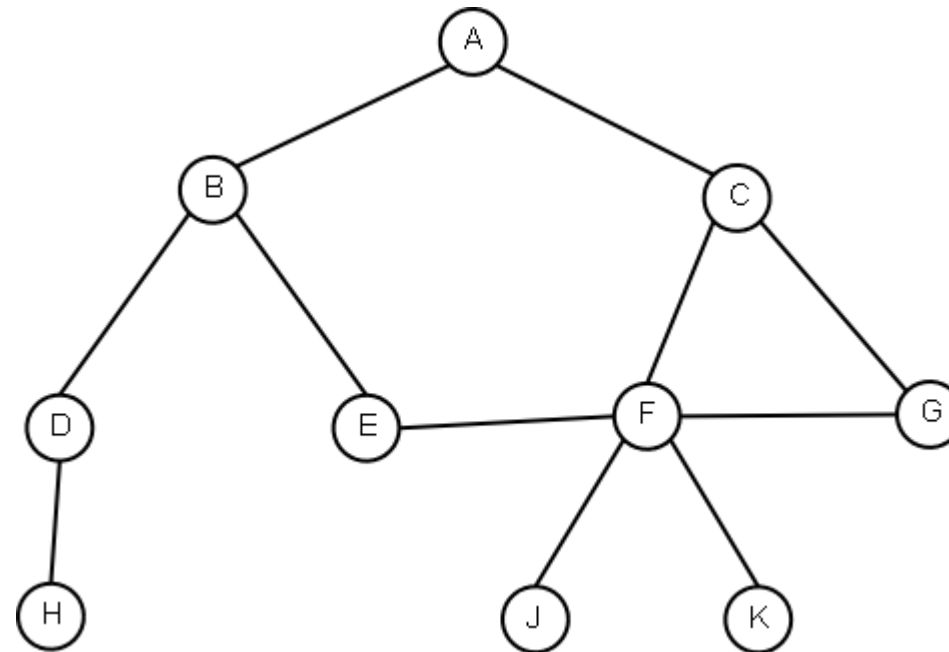
# Übung Breitendurchlauf

- Siehe Skript Aufgabe 6



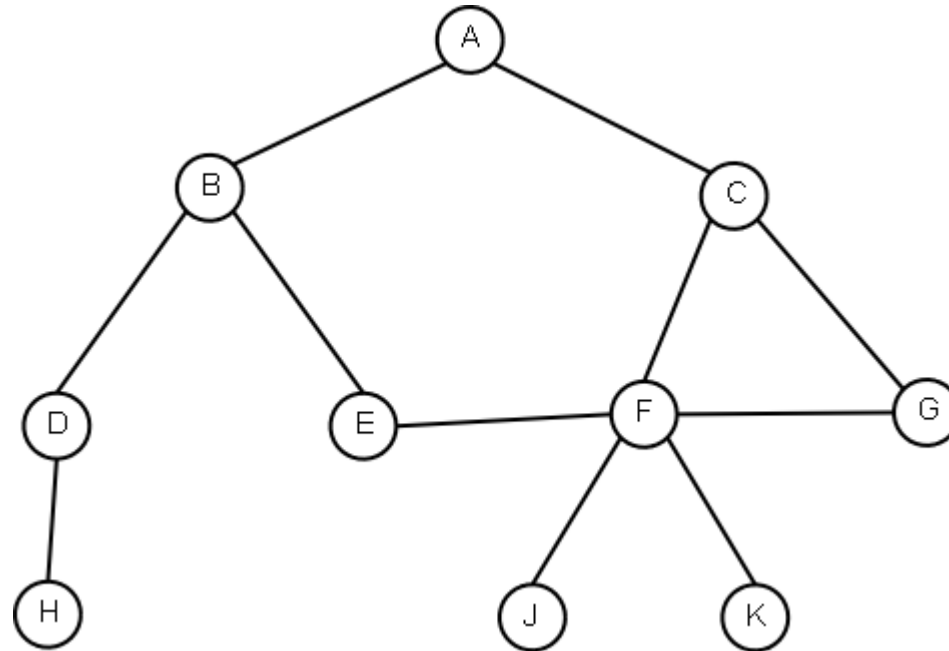
# Übung Breitendurchlauf

- Siehe Skript Aufgabe 6



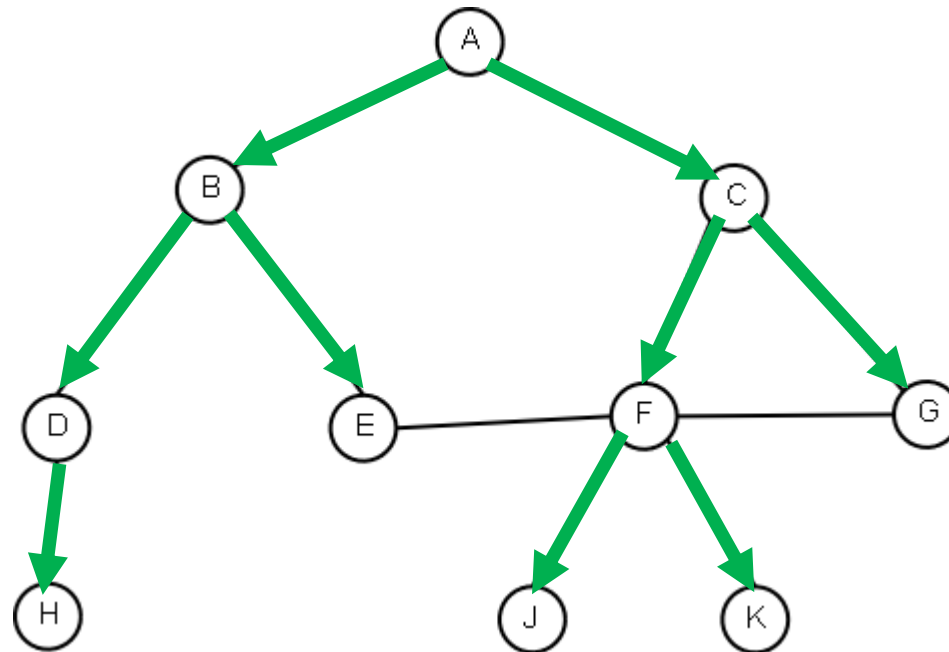
# Übung Breitendurchlauf

- Lösung: A – B – C – D – E – F – G – H – J – K



# Übung Breitendurchlauf

- Lösung: A – B – C – D – E – F – G – H – J – K



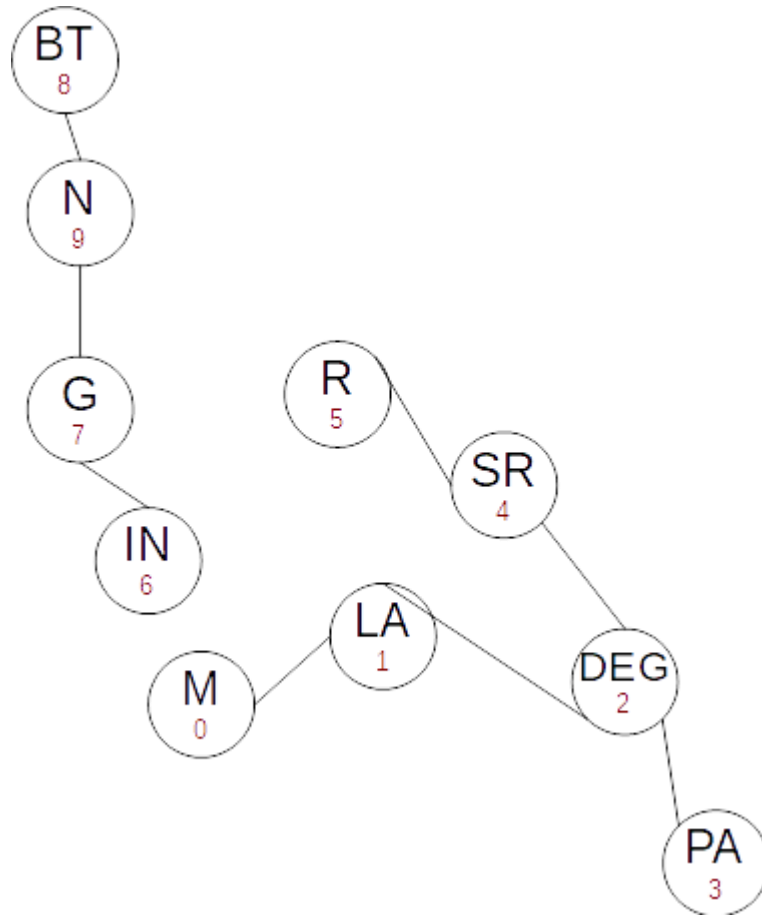


# Fehlerquellen – Breitensuche



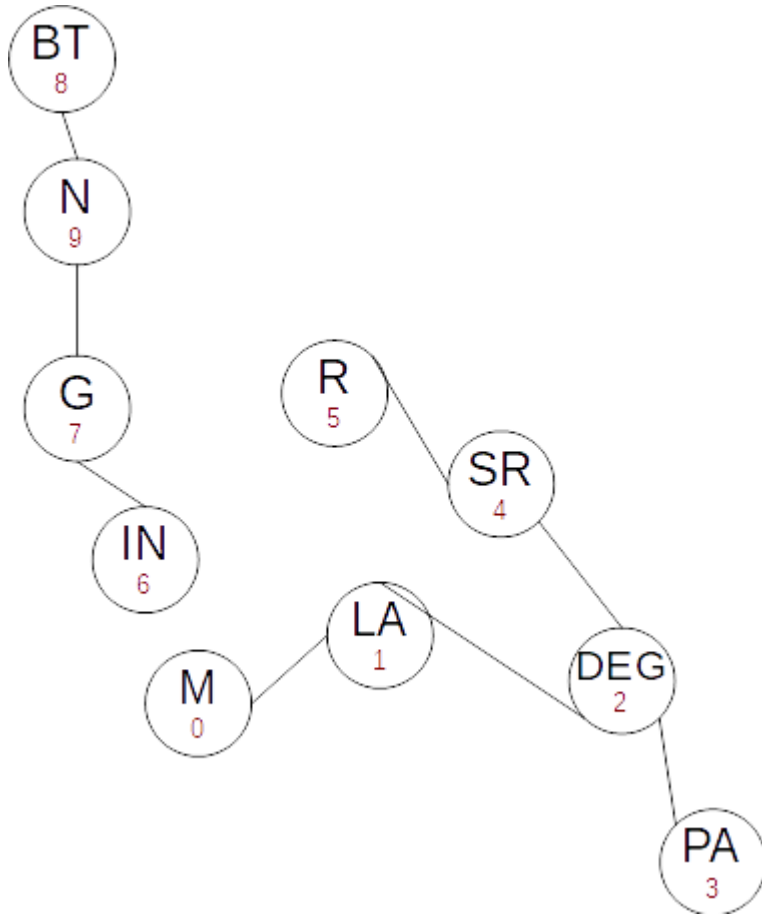


# Breitendurchlauf



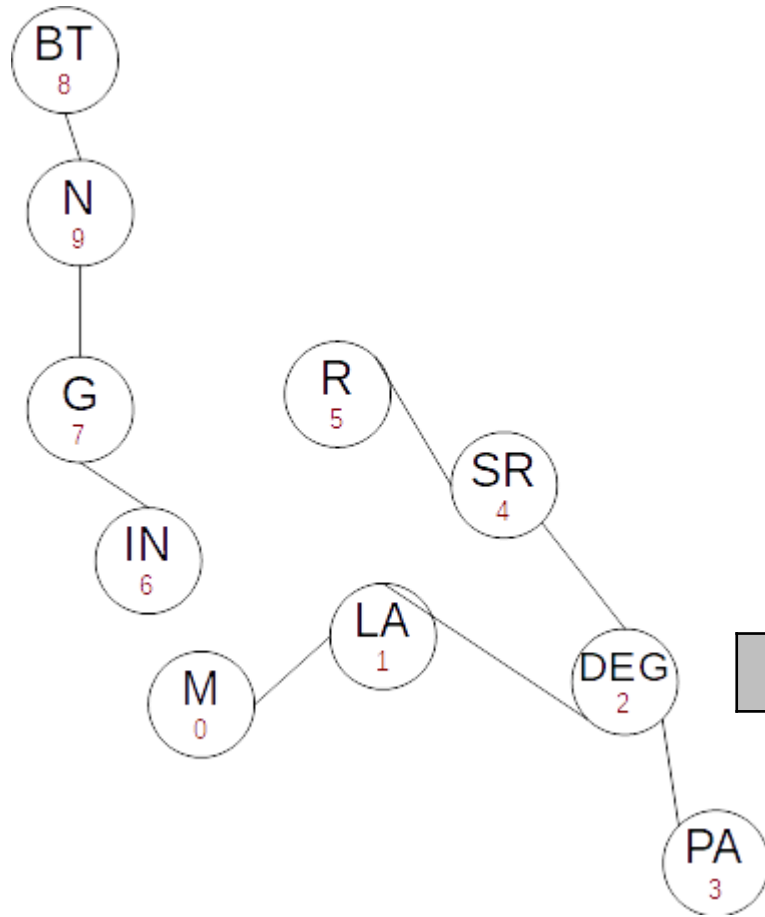
- Hier eine Übersicht eines des Testgraphen.
- Jeder Knoten wird mit seiner Index-Nummer angezeigt.

# Breitendurchlauf



- Hier eine Übersicht eines des Testgraphen.
  - Jeder Knoten wird mit seiner Index-Nummer angezeigt.
  - Es fällt auf: Der Graph ist **nicht zusammenhängend**.
- Hier liegt das Problem!

# Breitendurchlauf



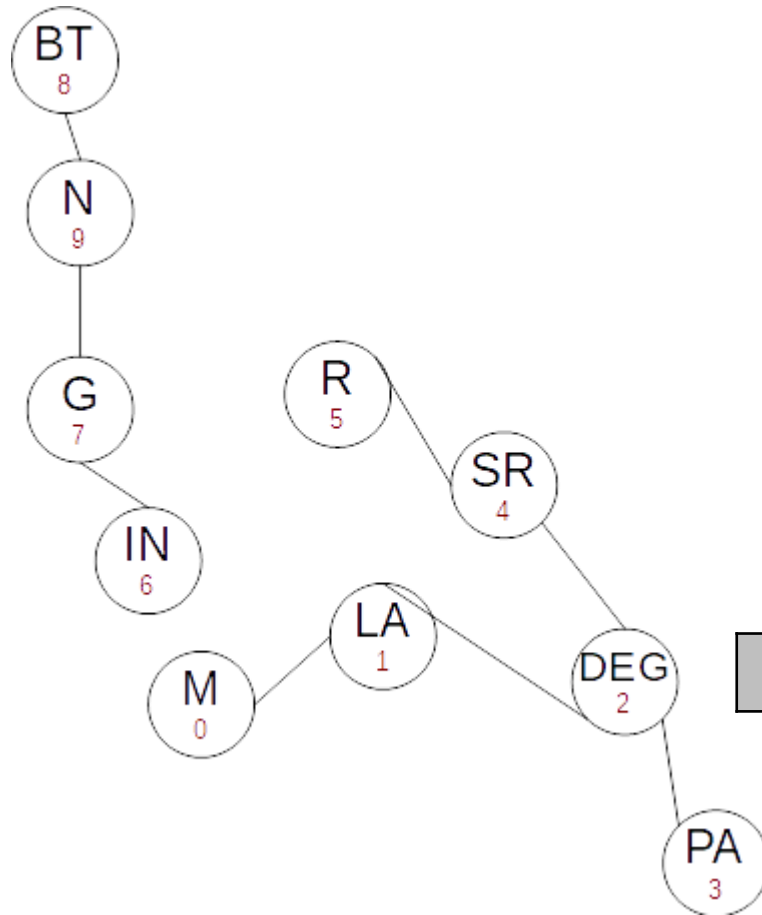
Beispiel:

Breitendurchlauf, Beginn Landshut

Merke-Array:

1	0	2	3	4	5				
---	---	---	---	---	---	--	--	--	--

# Breitendurchlauf



Beispiel:

Breitendurchlauf, Beginn Landshut

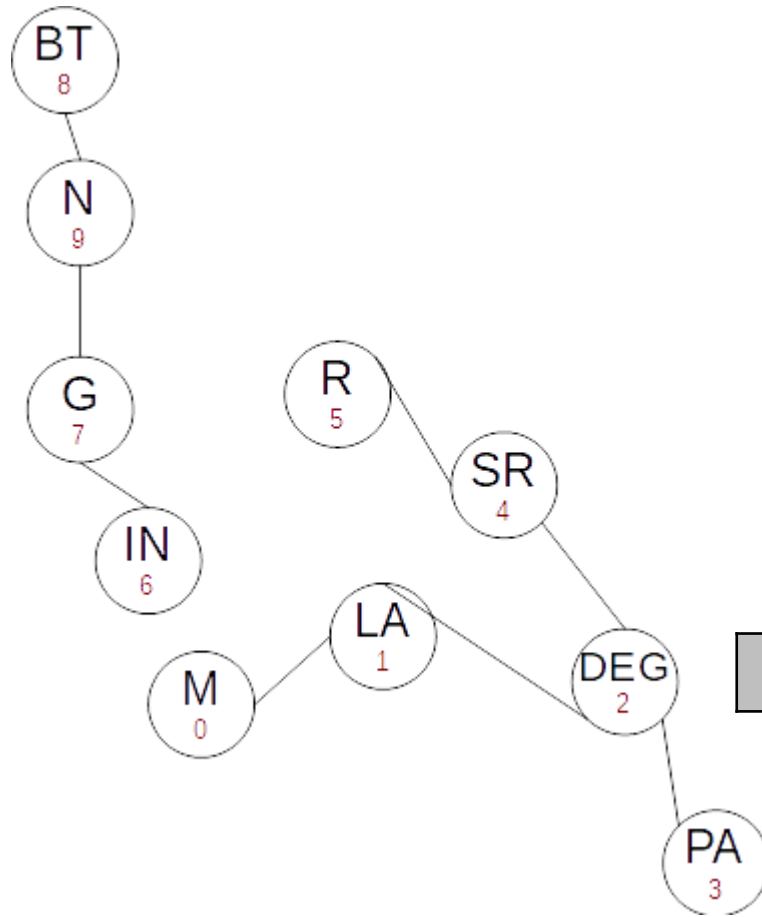
Merke-Array:

1	0	2	3	4	5				
---	---	---	---	---	---	--	--	--	--



Der Breitendurchlauf endet bei R (Nummer 5), weil der Graph nicht zusammenhängend ist.

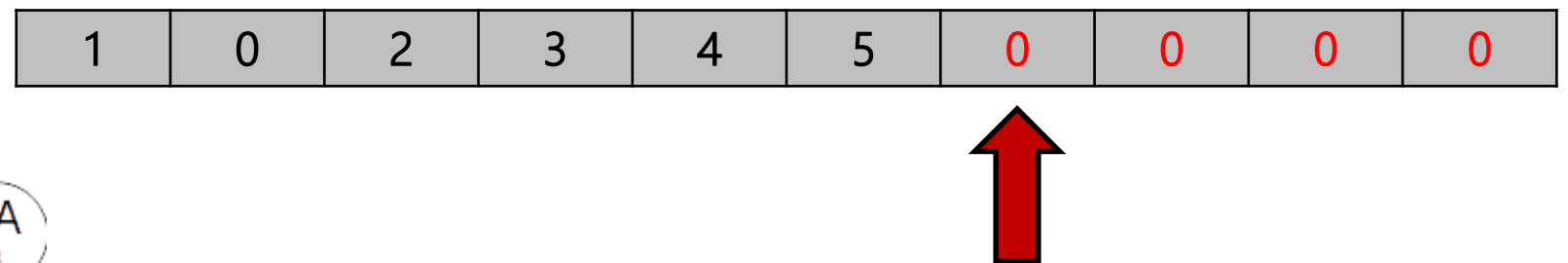
# Breitendurchlauf



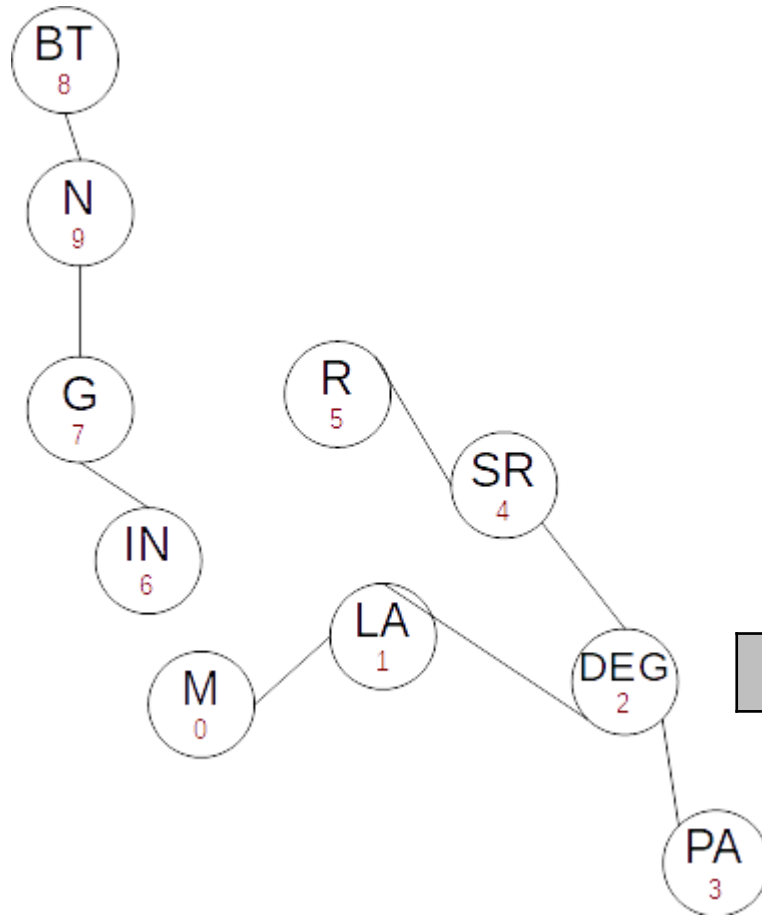
## Problem:

Java initialisiert Integer, sprich unser Schlangen-Array, standardmäßig mit 0-ern.

Merke-Array:

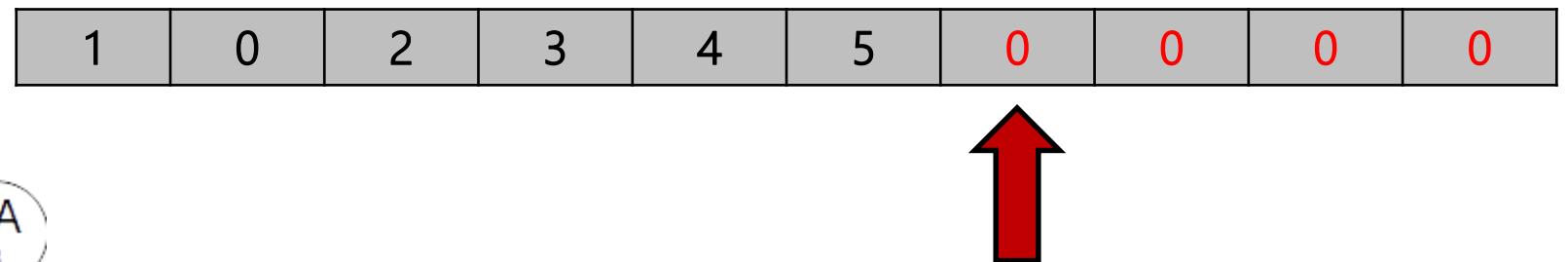


# Breitendurchlauf

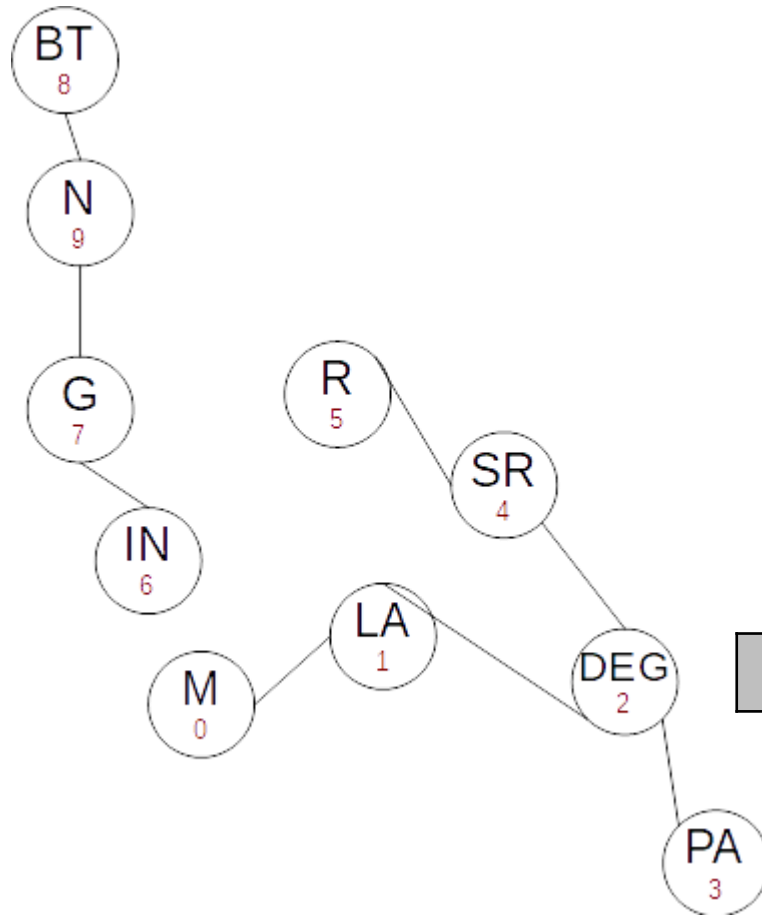


0 ist aber der Index von München!  
Deshalb wird noch 4 mal München  
auf fertig gesetzt und das Ganze  
ausgegeben. ;-)

Merke-Array:



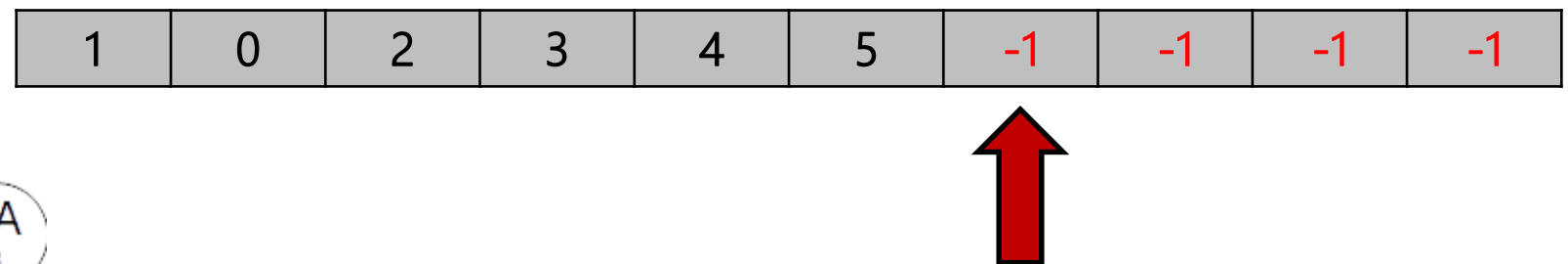
# Breitendurchlauf



**Lösung** des Problems:

Das Array wird mit -1 vorinitialisiert und der Inhalt der for-Schleife nur ausgeführt, wenn der Arrayplatz  $\neq -1$  ist.

Merke-Array:



# Breitendurchlauf

- Eine wesentlich elegantere Lösung ist natürlich eine Umsetzung mit einer dynamischen Merke-Schlange.
- z.B. mit der Java ArrayList.
- Die von Java vorgegebene ArrayList ist von der Datenstruktur wie eine Liste aufgebaut, die aber auch wie ein Array mit Indizes verwendet werden kann!
- Siehe JAVA-API