

# *Kapitel 3.3: Zusammenhangskomponenten*

*Algorithmen und Datenstrukturen  
WS 2023/24*

**Prof. Dr. Sándor Fekete**

### DEFINITION 3.6 (Wald, Baum)

- (1) Ein Wald ist ein kreisfreier Graph.
- (2) Ein Baum ist eine Zusammenhangskomponente in einem Wald.  
(Also: ein kreisfreier, zusammenhängender Graph)
- (3) Ein aufspannender Baum ist ein Baum, der alle Knoten verbindet. (Manchmal auch: Spannbaum. Englisch: "spanning tree")

# Algorithmus 3.7

INPUT: Graph  $G = (V, E)$ , Knoten  $s$

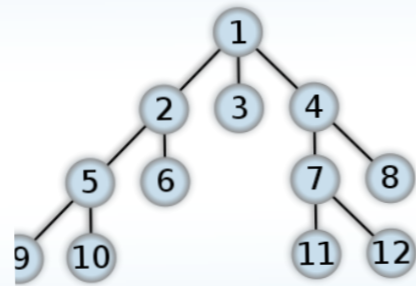
OUTPUT: Knotenmenge  $Y \subseteq V$ , die von  $s$  aus erreichbar ist,  
Kantenmenge  $T \subseteq E$ , die die Erreichbarkeit sicherstellt

1. Sei  $R := \{s\}$ ,  $Y := \{s\}$ ,  $T := \emptyset$
2. WHILE ( $R \neq \emptyset$ ) DO {
  - 2.1. Wähle  $v \in R$
  - 2.2. IF (es gibt kein  $w \in V \setminus Y$  mit  $e = \{v, w\} \in E$ ) THEN
    - 2.2.1.  $R := R \setminus \{v\}$
  - 2.3. ELSE {
    - 2.3.1. Wähle ein  $w \in V \setminus Y$  mit  $e = \{v, w\} \in E$
    - 2.3.2. Setze  $R := R \cup \{w\}$ ,  $Y := Y \cup \{w\}$ ,  $T := T \cup \{e\}$}}
3. STOP

**Satz 3.8.** *Der Algorithmus 3.7 ist*

*(1) endlich*

*(2) korrekt.*



*Kapitel 3.4:*  
*Warteschlange und Stapel*  
*Algorithmen und Datenstrukturen*  
*WS 2023/24*

**Prof. Dr. Sándor Fekete**

# Algorithmus 3.7

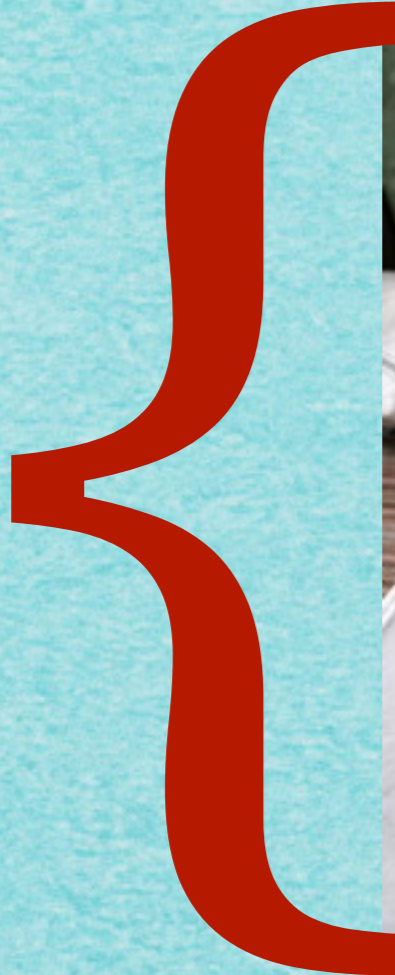
INPUT: Graph  $G = (V, E)$ , Knoten  $s$

OUTPUT: Knotenmenge  $Y \subseteq V$ , die von  $s$  aus erreichbar ist,  
Kantenmenge  $T \subseteq E$ , die die Erreichbarkeit sicherstellt

1. Sei  $R := \{s\}$ ,  $Y := \{s\}$ ,  $T := \emptyset$
2. WHILE ( $R \neq \emptyset$ ) DO {
  - 2.1. Wähle  $v \in R$
  - 2.2. IF (es gibt kein  $w \in V \setminus Y$  mit  $e = \{v, w\} \in E$ ) THEN
    - 2.2.1.  $R := R \setminus \{v\}$
  - 2.3. ELSE {
    - 2.3.1. Wähle ein  $w \in V \setminus Y$  mit  $e = \{v, w\} \in E$
    - 2.3.2. Setze  $R := R \cup \{w\}$ ,  $Y := Y \cup \{w\}$ ,  $T := T \cup \{e\}$}
3. STOP

# Wie verwalten wir R?

R



Wähle v!

Waschkorb



**A WEBCOMIC OF ROMANCE,  
SARCASM, MATH, AND LANGUAGE.**



HOME ORGANIZATION TIP:  
JUST GIVE UP.



# Algorithmus 3.7

INPUT: Graph  $G = (V, E)$ , Knoten  $s$

OUTPUT: Knotenmenge  $Y \subseteq V$ , die von  $s$  aus erreichbar ist,  
Kantenmenge  $T \subseteq E$ , die die Erreichbarkeit sicherstellt

1. Sei  $R := \{s\}$ ,  $Y := \{s\}$ ,  $T := \emptyset$
2. WHILE ( $R \neq \emptyset$ ) DO {
  - 2.1. Wähle  $v \in R$
  - 2.2. IF (es gibt kein  $w \in V \setminus Y$  mit  $e = \{v, w\} \in E$ ) THEN
    - 2.2.1.  $R := R \setminus \{v\}$
  - 2.3. ELSE {
    - 2.3.1. Wähle ein  $w \in V \setminus Y$  mit  $e = \{v, w\} \in E$
    - 2.3.2. Setze  $R := R \cup \{w\}$ ,  $Y := Y \cup \{w\}$ ,  $T := T \cup \{e\}$}
3. STOP

# A & D

I get the job done.  
What the hell do you  
want?

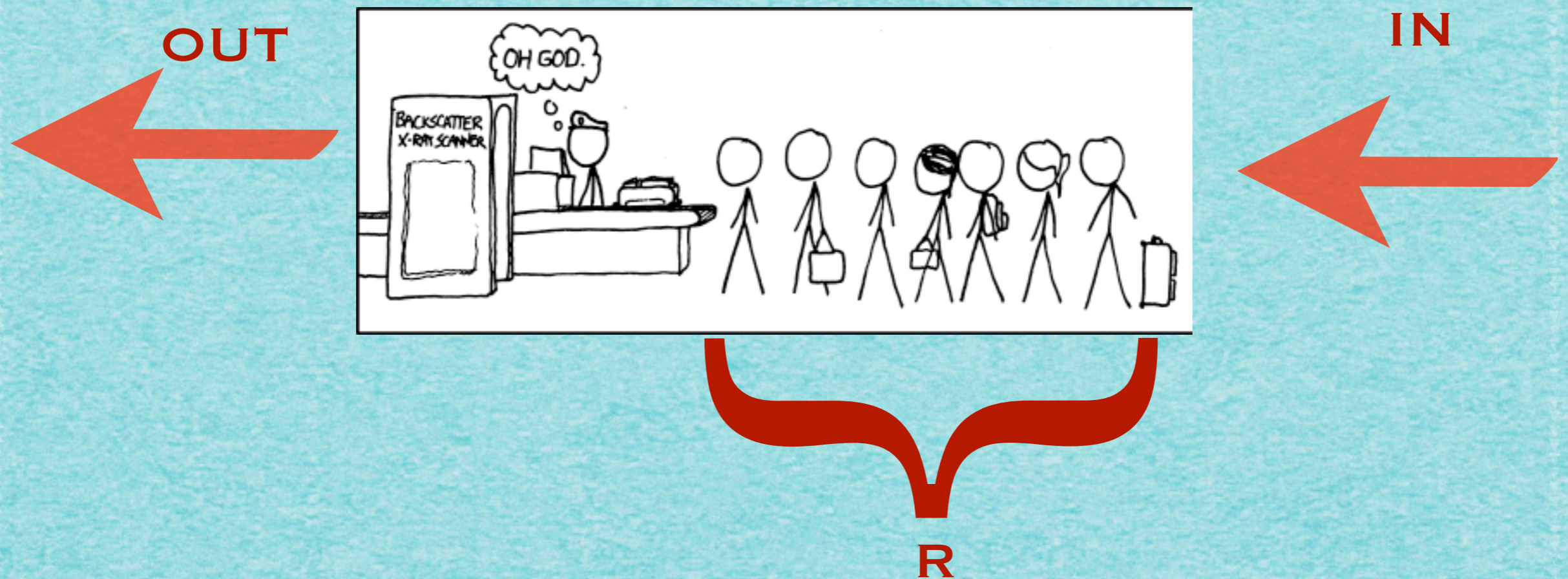
CAN YOU MAKE IT  
WITHOUT KILLING  
YOURSELF?



**Algorithmus**

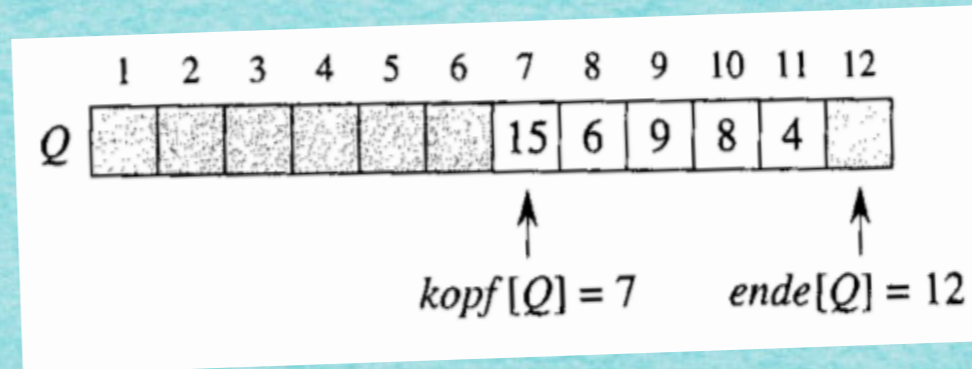
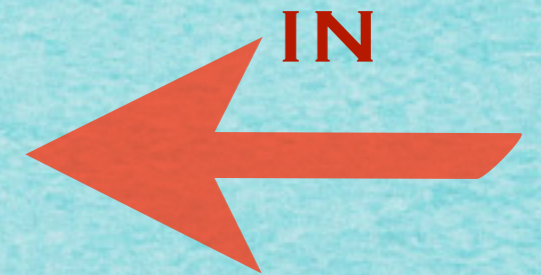
**DATENSTRUKTUR**

# DATENSTRUKTUR I

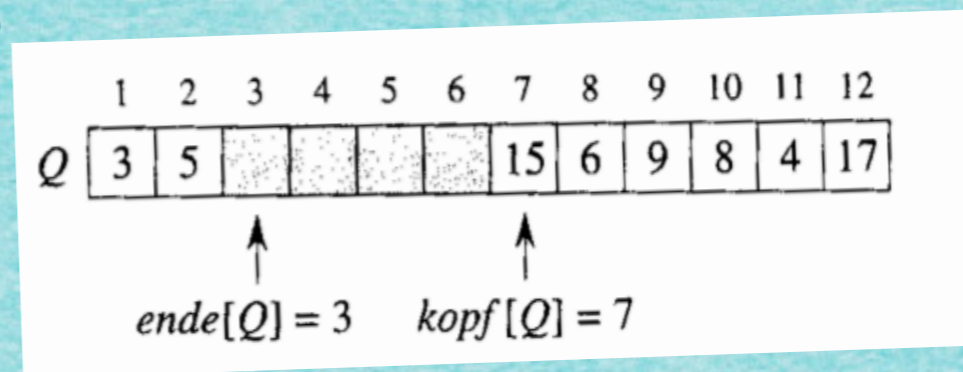


**WARTESCHLANGE: FIRST IN - FIRST OUT**

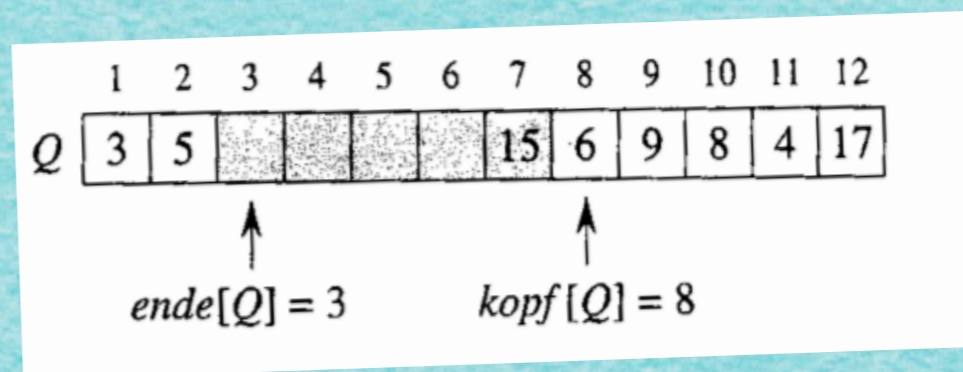
# WARTESCHLANGE AUF ARRAY UMGESETZT



ENQUEUE: 17, 3, 5

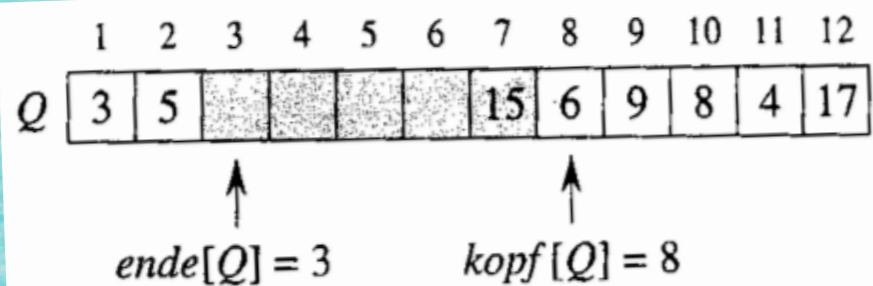
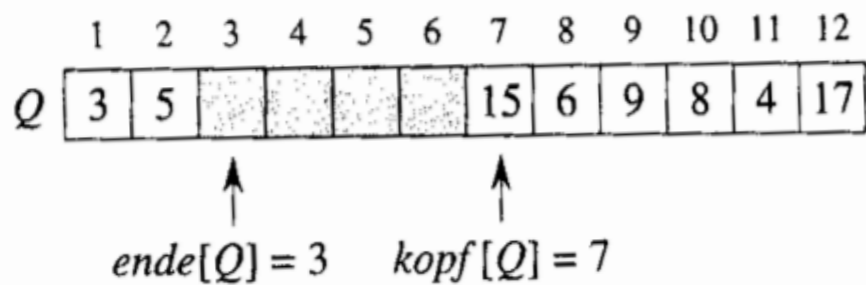
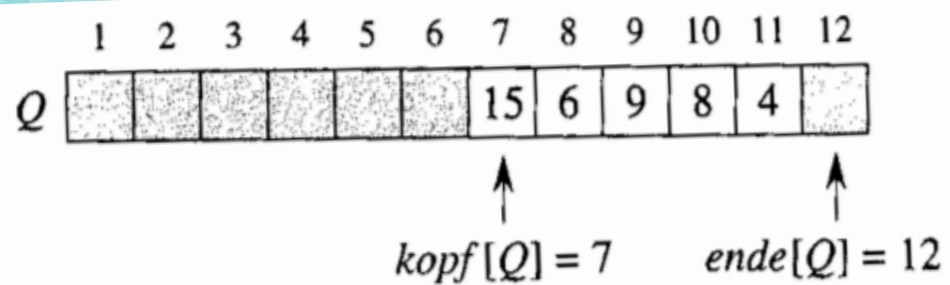


DEQUEUE:



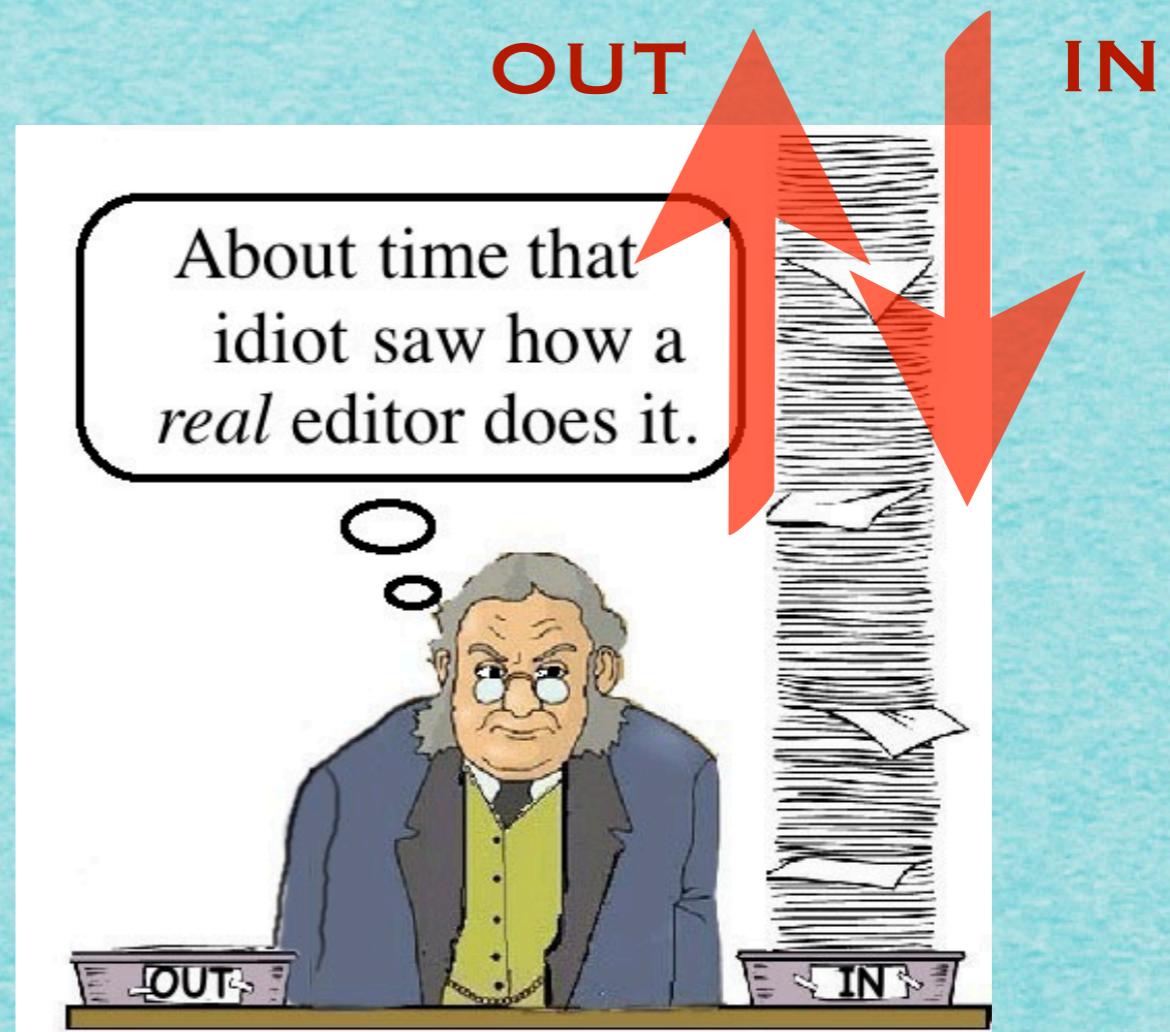
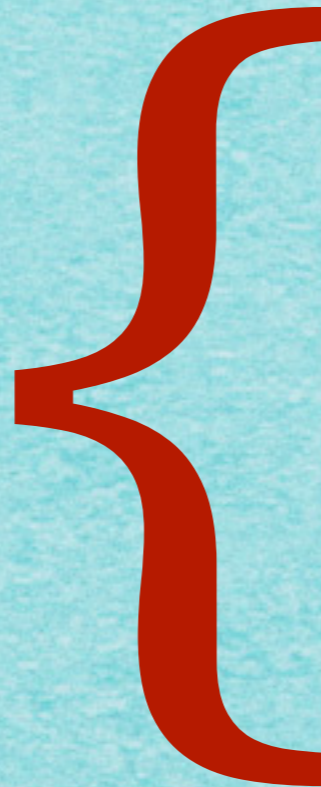


# WARTESCHLANGE AUF ARRAY UMGESETZT



# DATENSTRUKTUR II

R



**STAPEL: LAST IN - FIRST OUT**

# DATENSTRUKTUR II



**STAPEL: LAST IN - FIRST OUT**

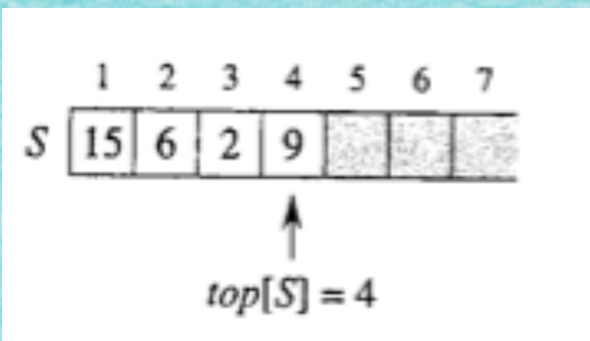


# DATENSTRUKTUR II

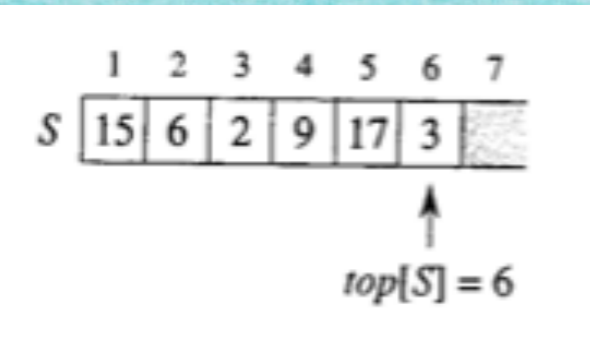


**STAPEL: LAST IN - FIRST OUT**

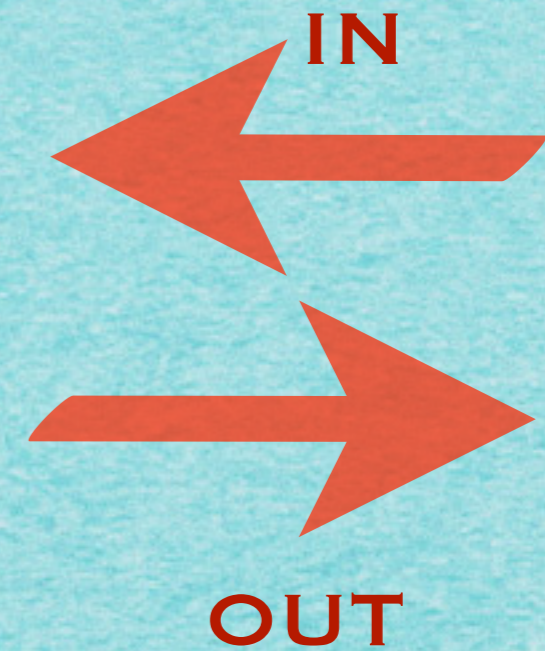
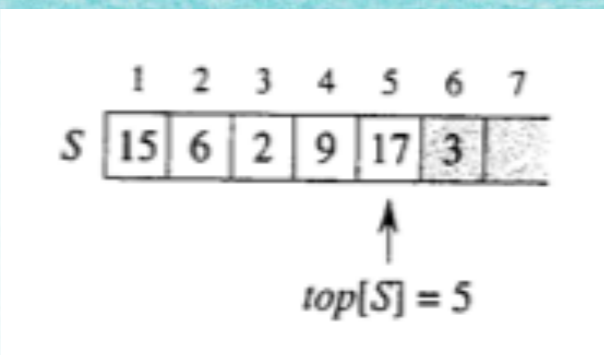
# STACK AUF ARRAY UMGESSETZT



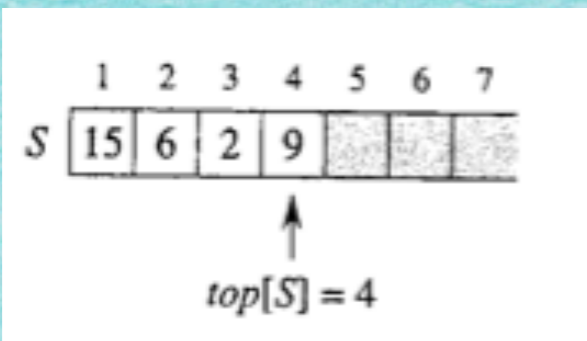
**PUSH: 17, 3**



**POP**

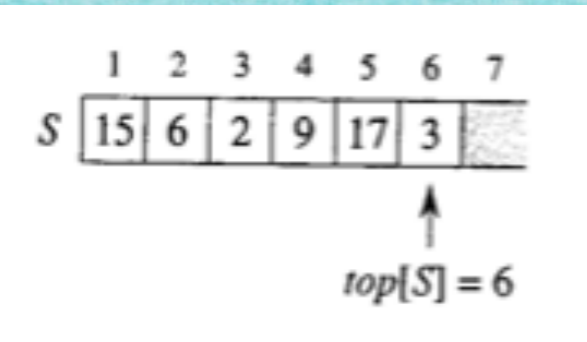


# STACK AUF ARRAY UMGESETZT



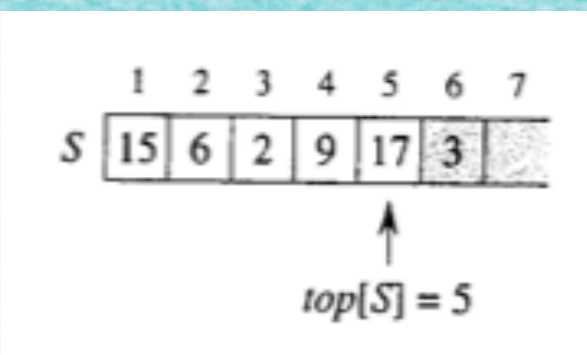
```

STACK-EMPTY(S)
1  if  $top[S] = 0$ 
2     then return WAHR
3     else return FALSCH
    
```



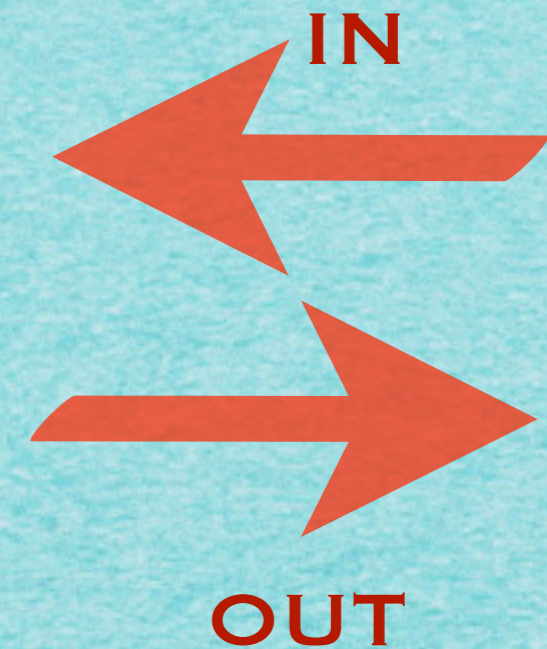
```

PUSH(S, x)
1   $top[S] \leftarrow top[S] + 1$ 
2   $S[top[S]] \leftarrow x$ 
    
```



```

POP(S)
1  if STACK-EMPTY(S)
2     then error "Unterlauf"
3     else  $top[S] \leftarrow top[S] - 1$ 
4           return  $S[top[S] + 1]$ 
    
```



# KARRIERECHANCEN!

## KARRIERE SPIEGEL

● Home ● Berufsstart ● Berufsleben ● Ausland ● Stellensuche ● Jobs

**Thema** Auto und Beruf - KarriereSPIEGEL »

Nachrichten > KarriereSPIEGEL > Berufsstart > Auto und Beruf - KarriereSPIEGEL > Auto-Industrie

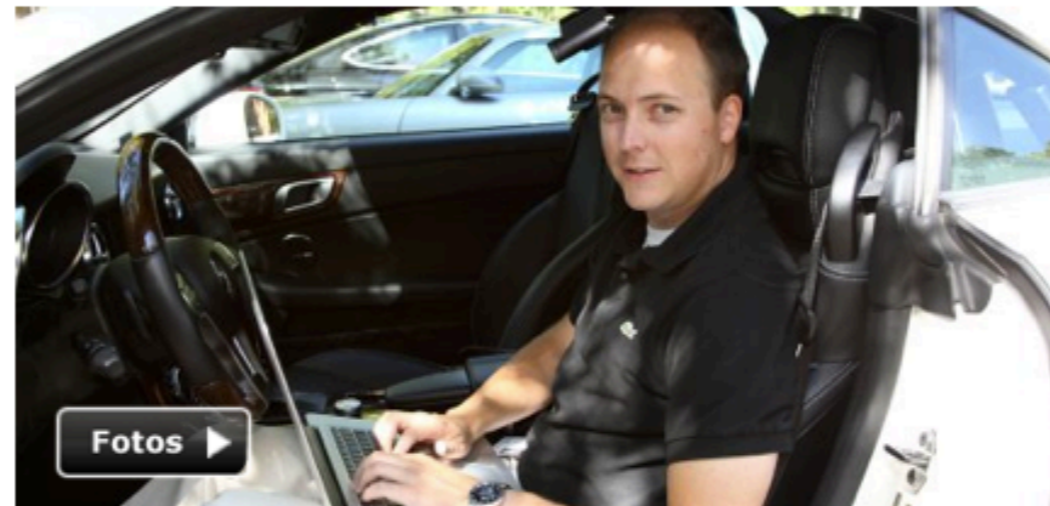
05.10.2012

Drucken | Senden | Feedback | Merken

Informatiker im Autobau

### Mehr Software als im Kampfjet

Von Peter Ilg



Fotos ▶

Daimler

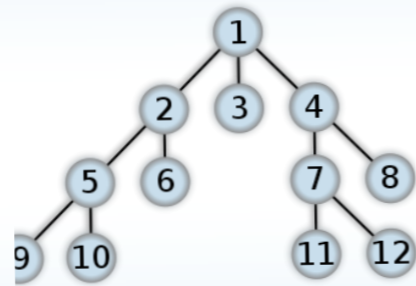
## KARRIERECHANCEN!

### **Am liebsten reine Informatiker**

Insgesamt 30 Mitarbeiter werkeln bei Daimler an der Systemarchitektur des Infotainment im Auto, die meisten sind Informatiker oder Ingenieure der Elektrotechnik. Vor sechs Jahren waren es noch halb so viele. "Bei Berufseinsteigern sind uns reine Informatiker am liebsten", sagt Matthias Stümpfle, der Leiter der Abteilung: "Sie sind in der Informatik gründlich ausgebildet, das Branchen-Know-how bringen wir ihnen bei."

# KARRIERECHANCEN!



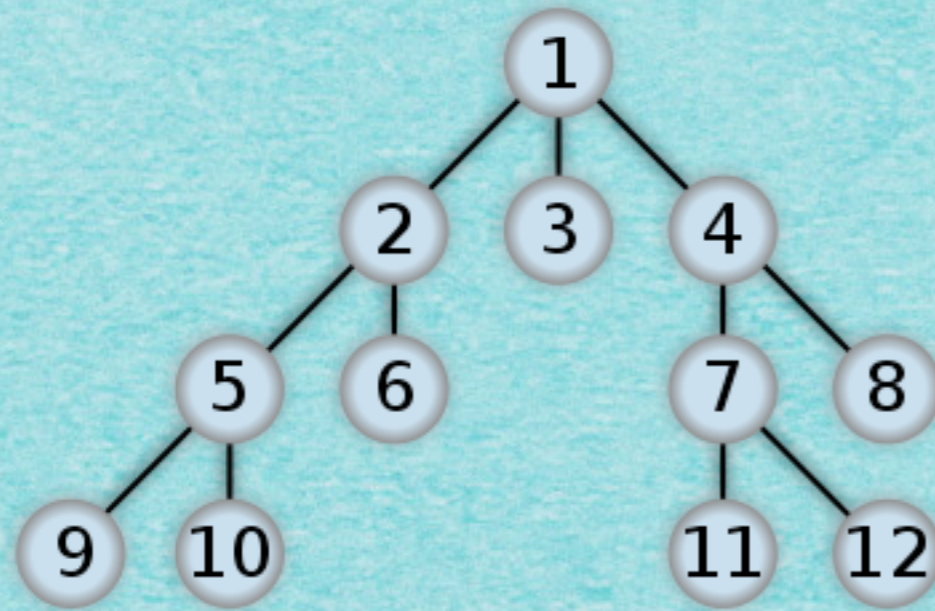


# *Kapitel 3.5: Tiefensuche und Breitensuche*

*Algorithmen und Datenstrukturen  
WS 2023/24*

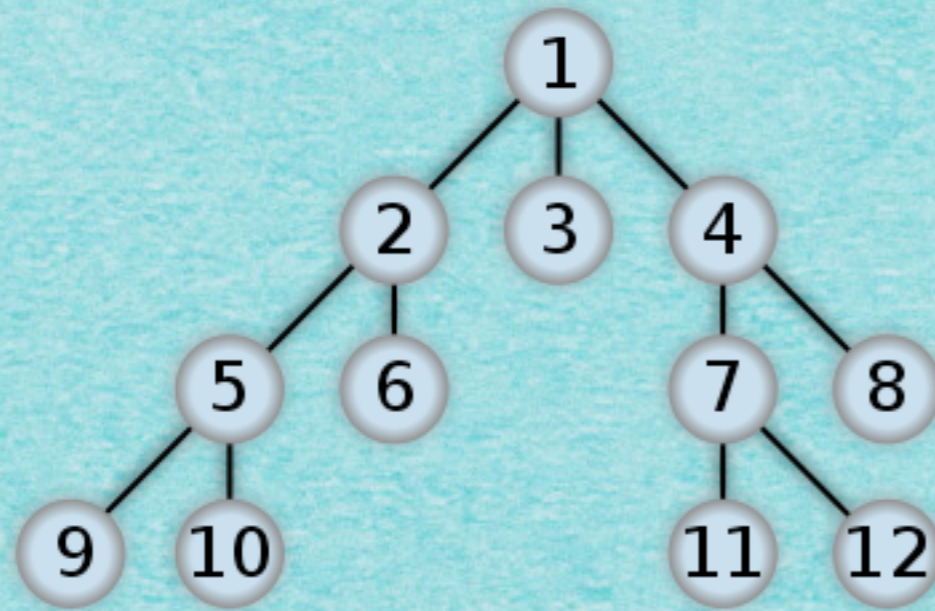
**Prof. Dr. Sándor Fekete**

# Graphenscan mit WARTESCHLANGE



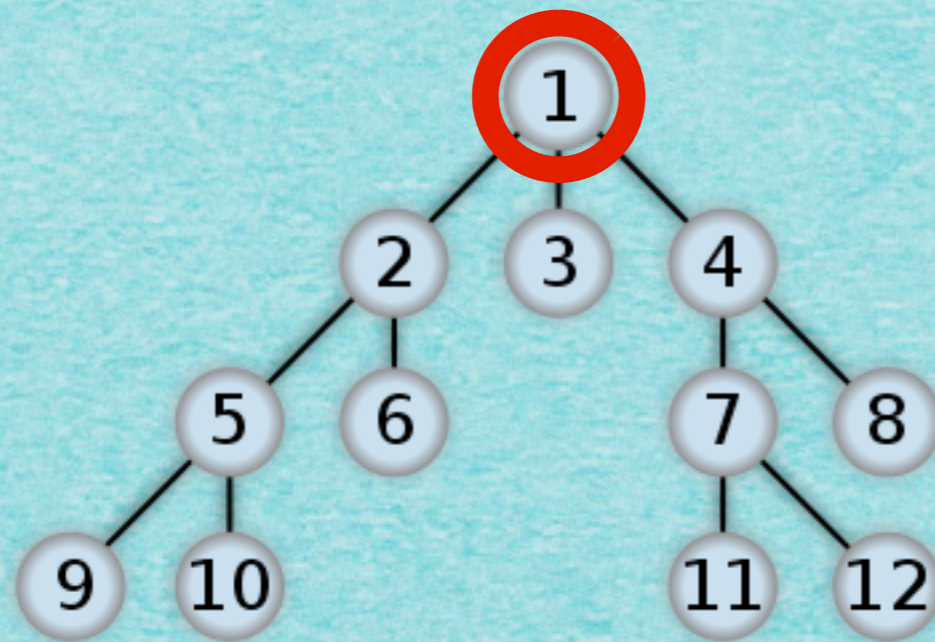


# Graphenscan mit WARTESCHLANGE



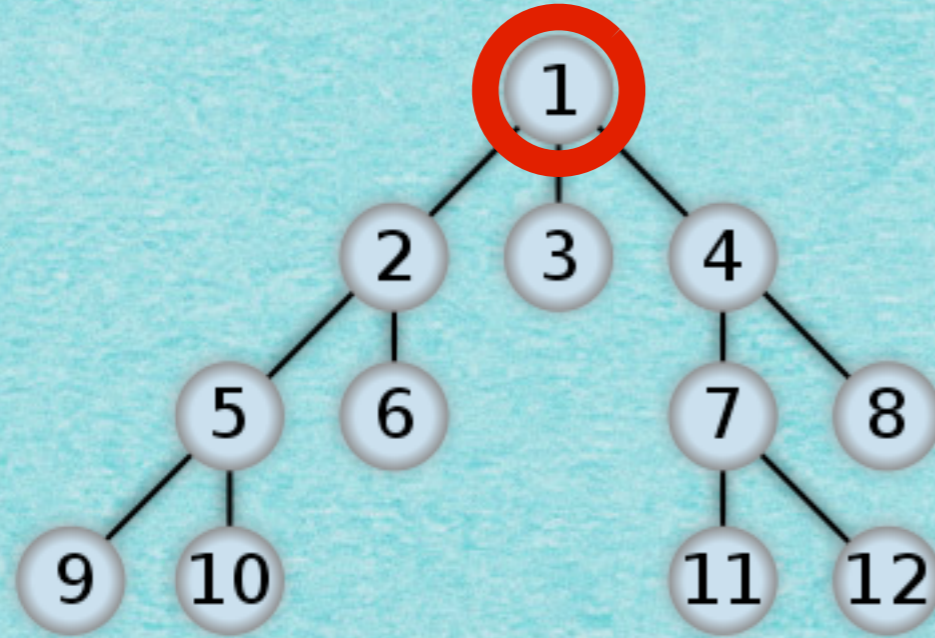
R:

# Graphenscan mit WARTESCHLANGE



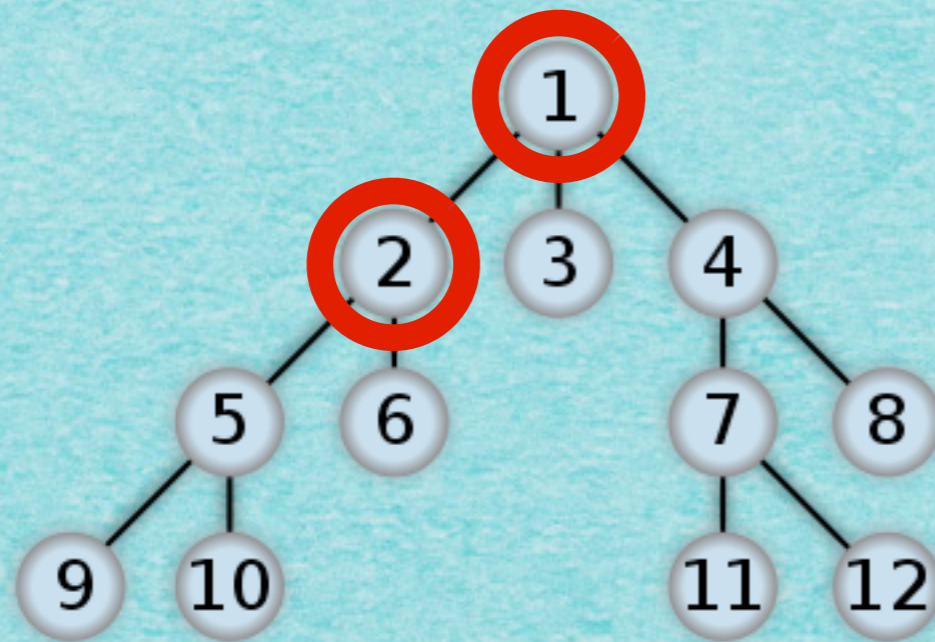
R:

# Graphenscan mit WARTESCHLANGE



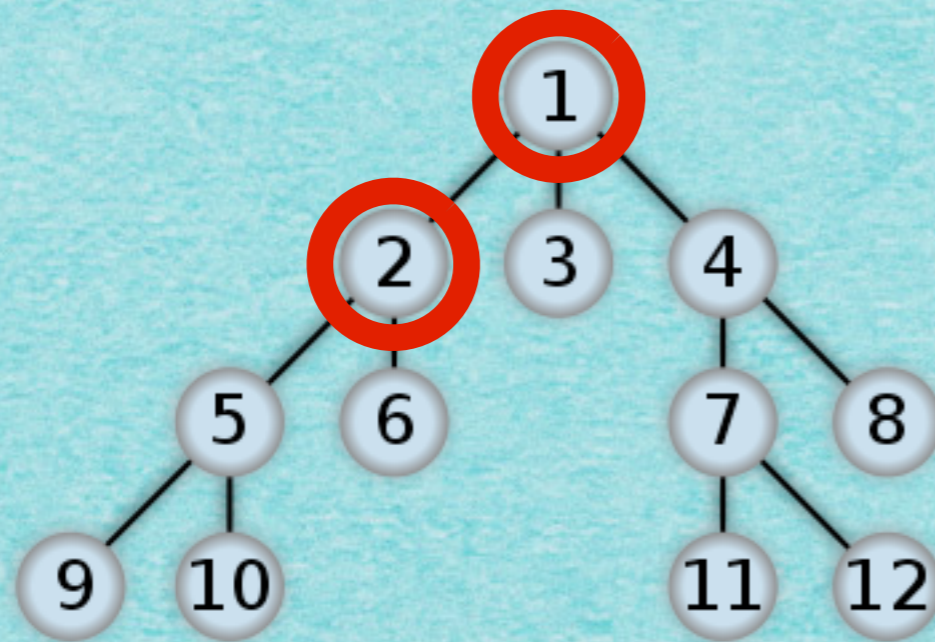
R: 1

# Graphenscan mit WARTESCHLANGE



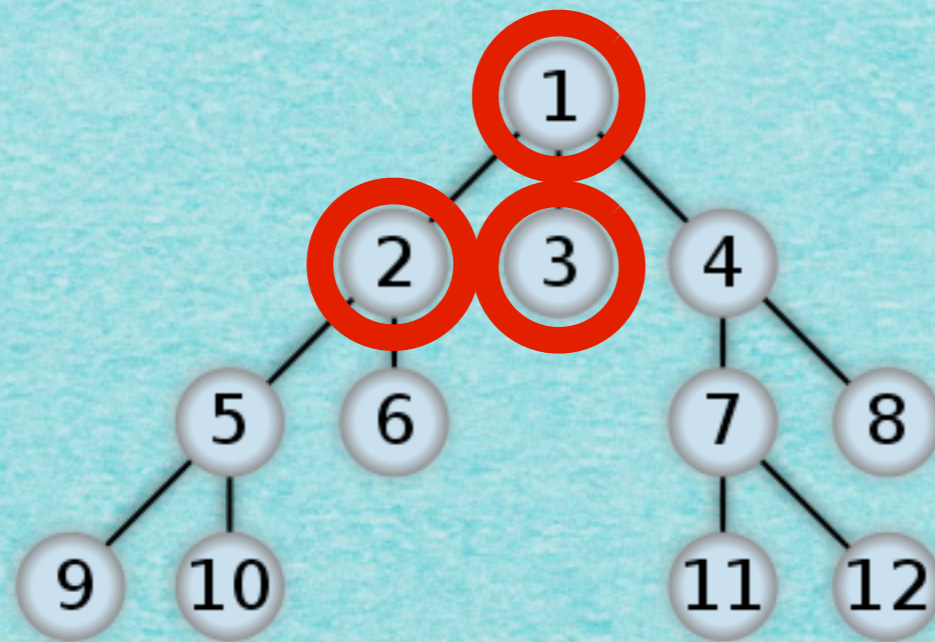
R: 1

# Graphenscan mit WARTESCHLANGE



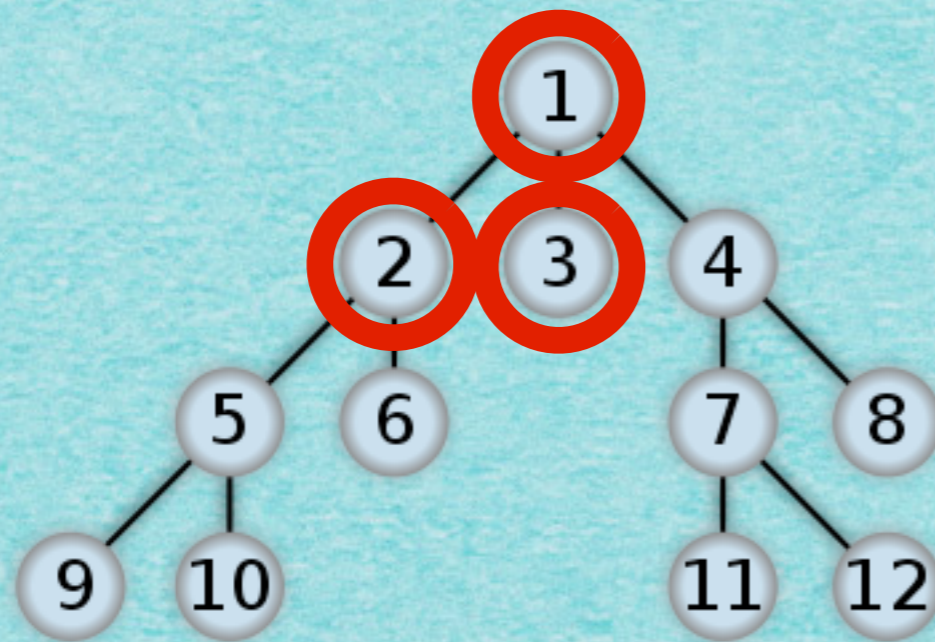
R: 1,2

# Graphenscan mit WARTESCHLANGE



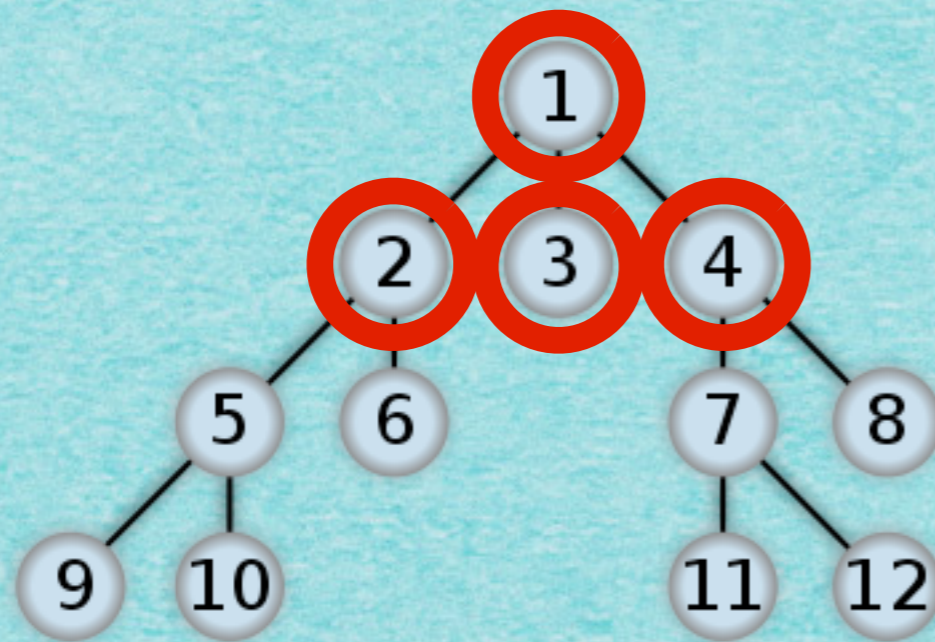
R: 1,2

# Graphenscan mit WARTESCHLANGE



R: 1,2,3

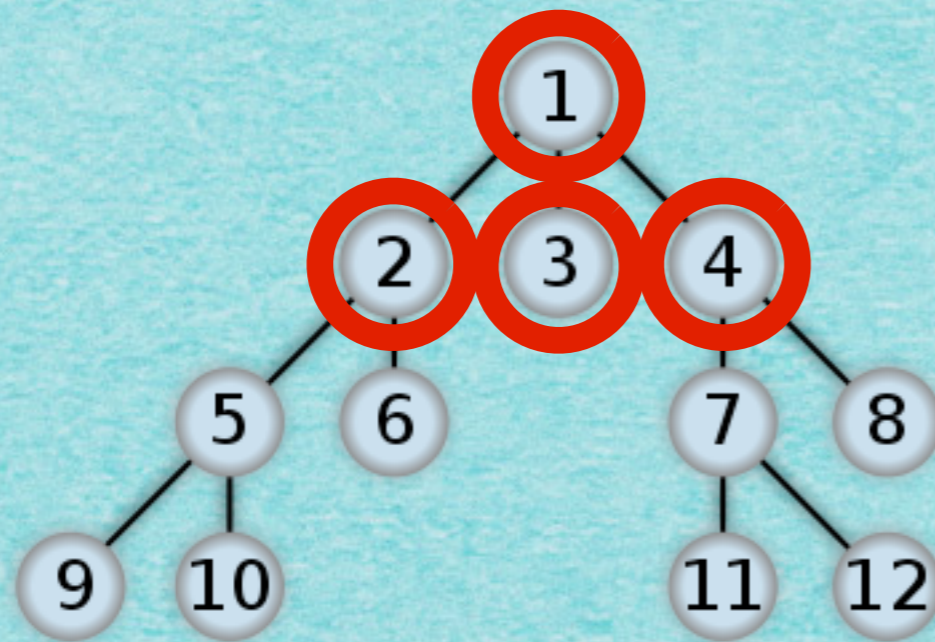
# Graphenscan mit WARTESCHLANGE



R: 1,2,3

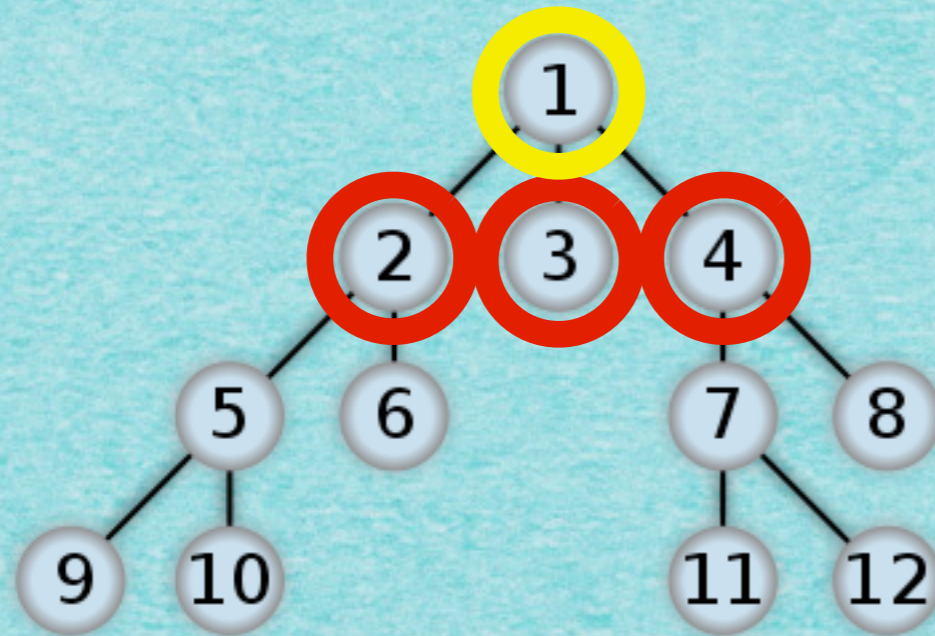


# Graphenscan mit WARTESCHLANGE



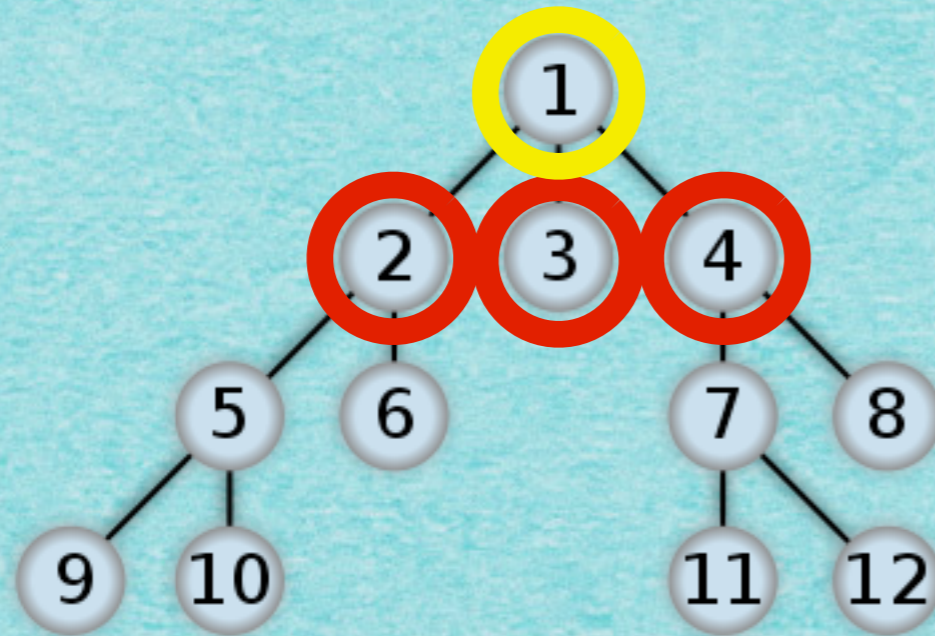
R: 1,2,3,4

# Graphenscan mit WARTESCHLANGE



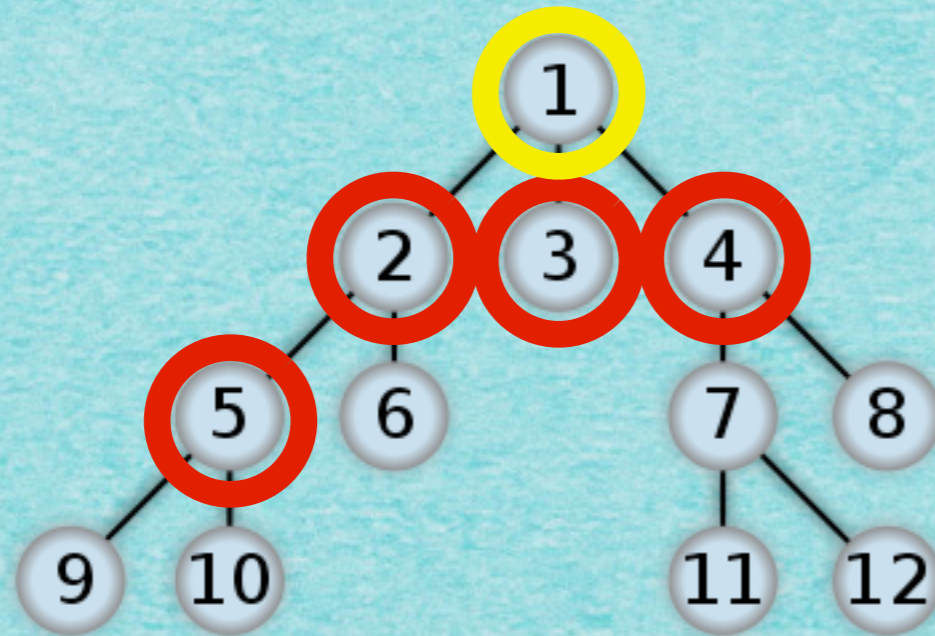
R: 1,2,3,4

# Graphenscan mit WARTESCHLANGE



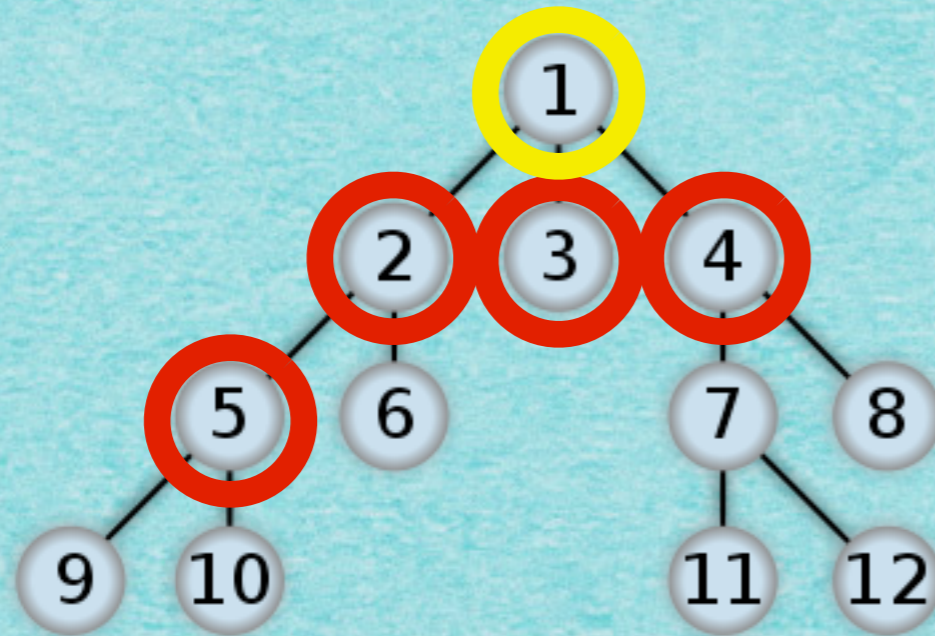
R: 2,3,4

# Graphenscan mit WARTESCHLANGE



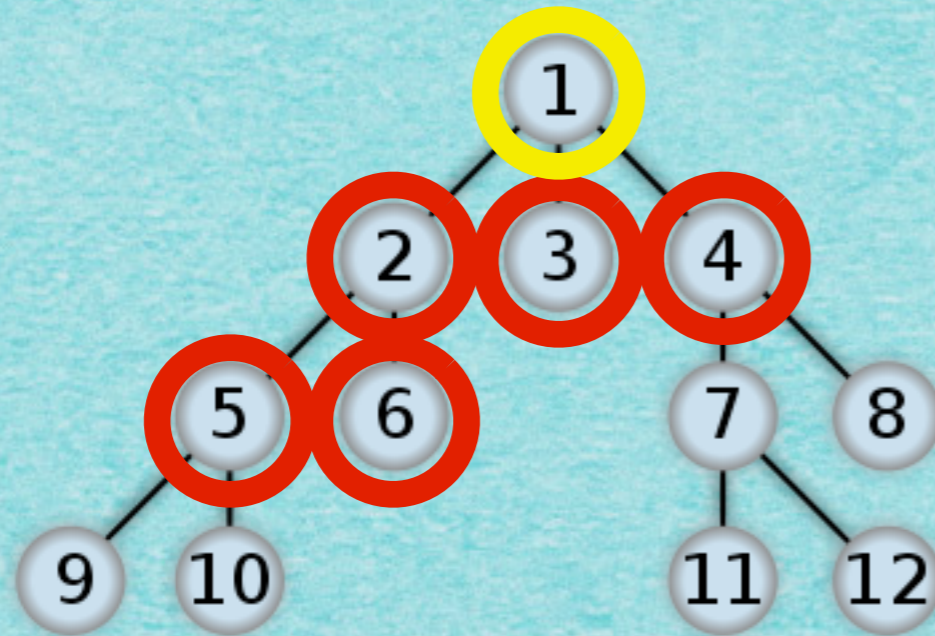
R: 2,3,4

# Graphenscan mit WARTESCHLANGE



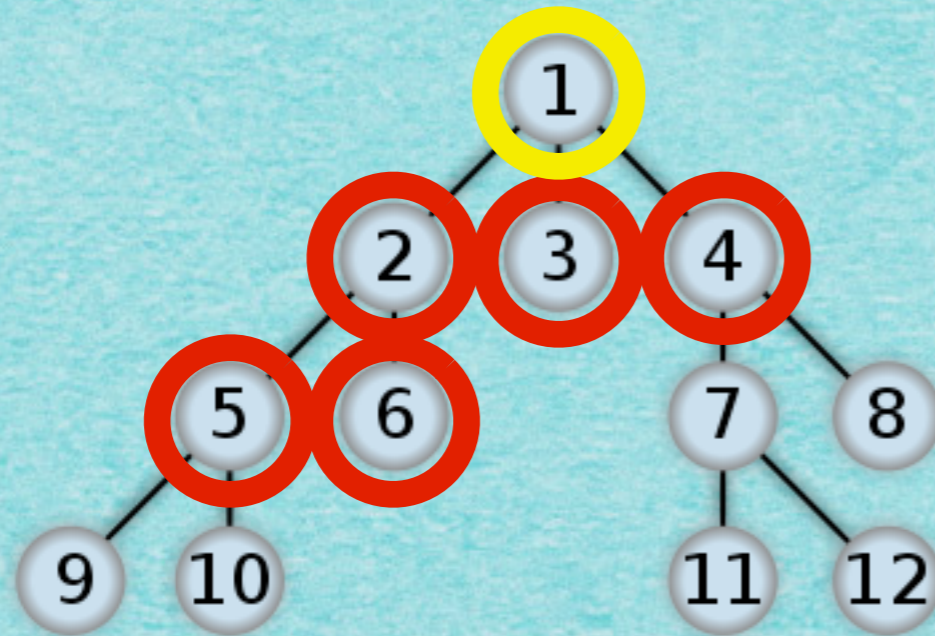
R: 2,3,4,5

# Graphenscan mit WARTESCHLANGE



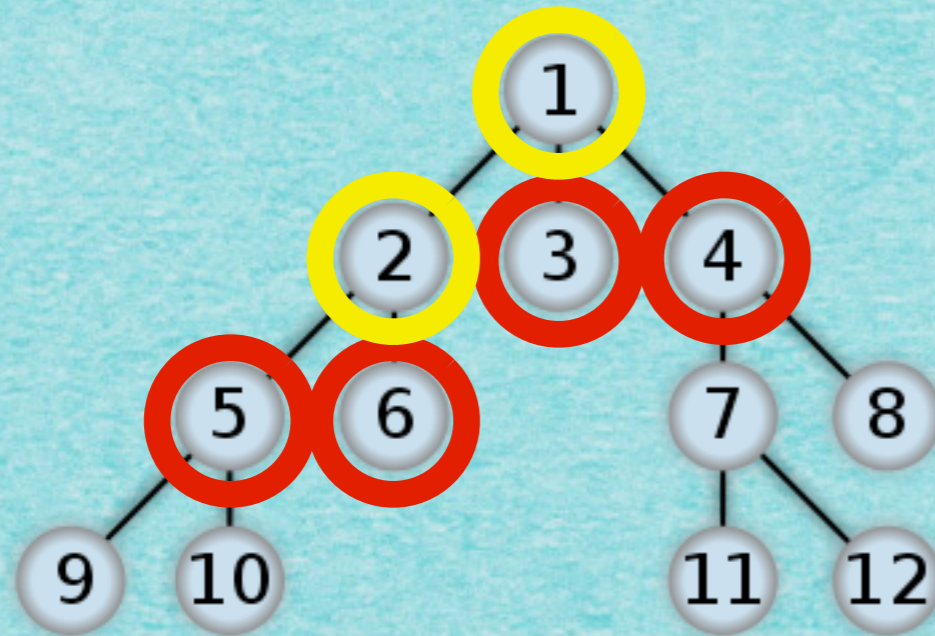
R: 2,3,4,5

# Graphenscan mit WARTESCHLANGE



R: 2,3,4,5,6

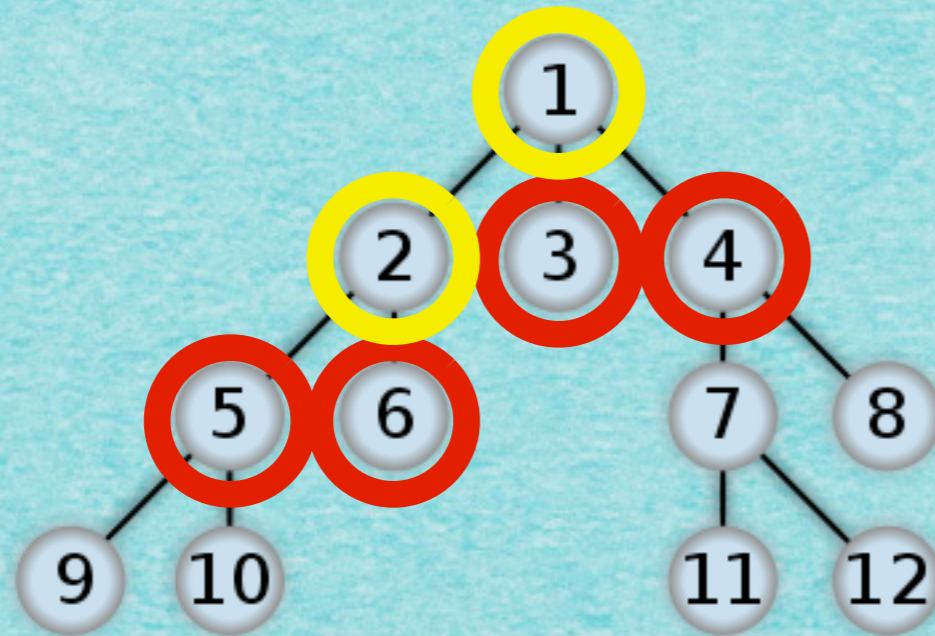
# Graphenscan mit WARTESCHLANGE



R: 2,3,4,5,6

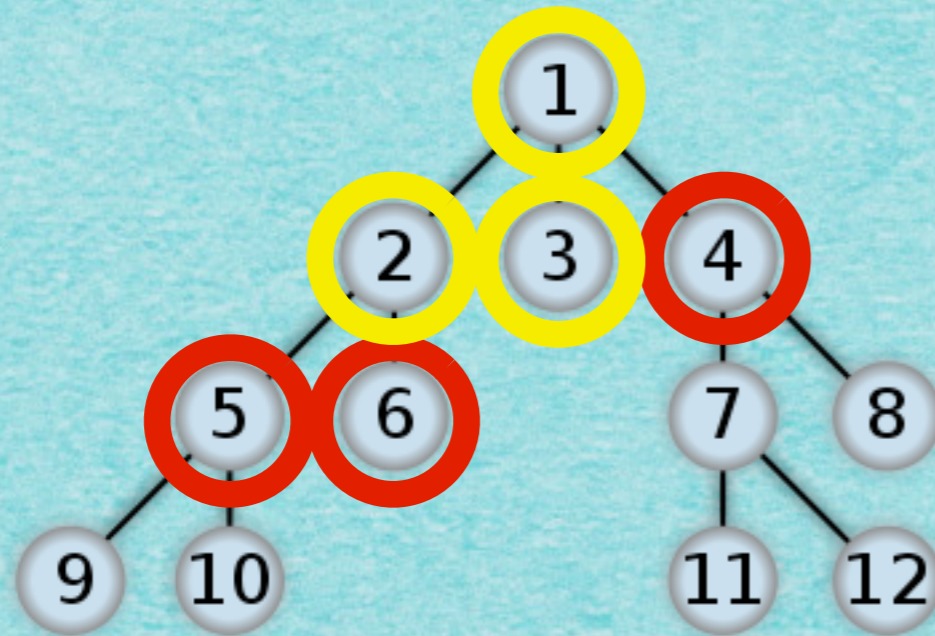


# Graphenscan mit WARTESCHLANGE



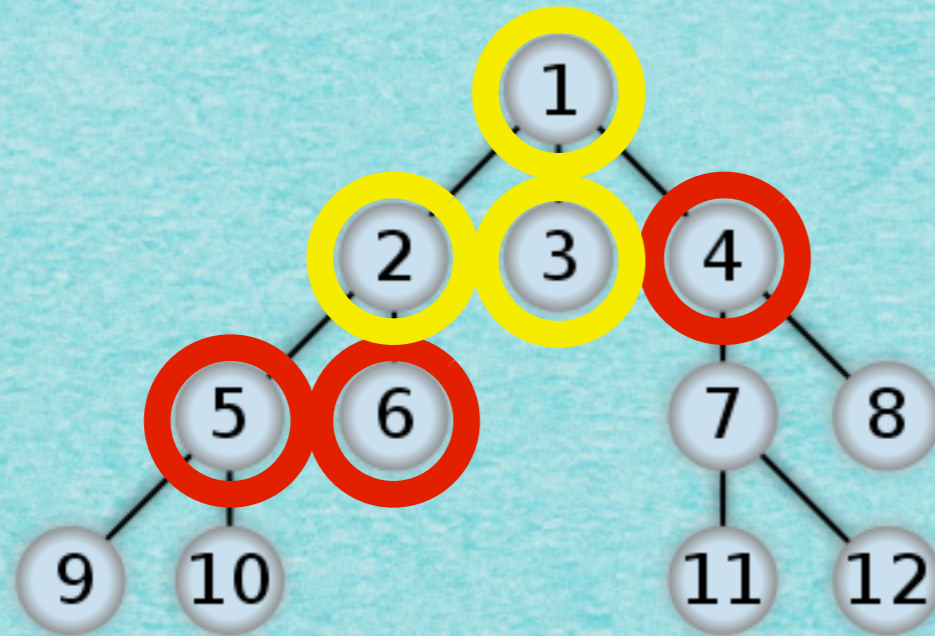
R: 3,4,5,6

# Graphenscan mit WARTESCHLANGE



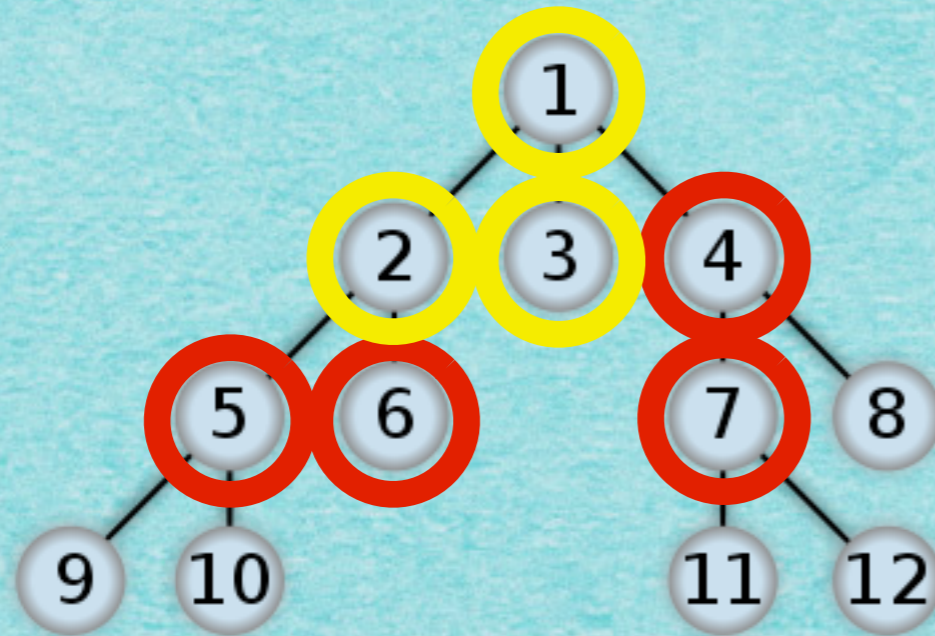
R: 3,4,5,6

# Graphenscan mit WARTESCHLANGE



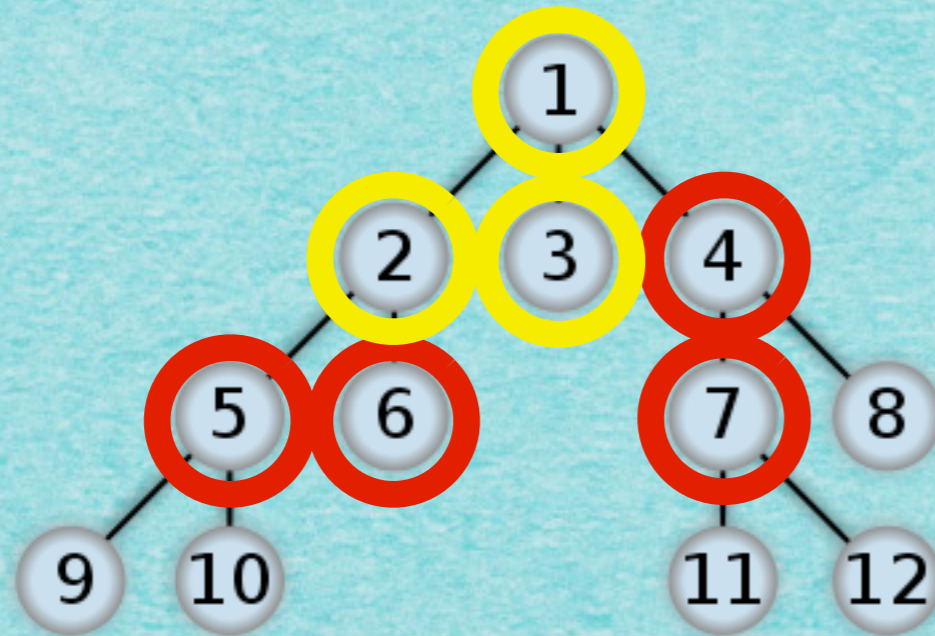
R: 4,5,6

# Graphenscan mit WARTESCHLANGE



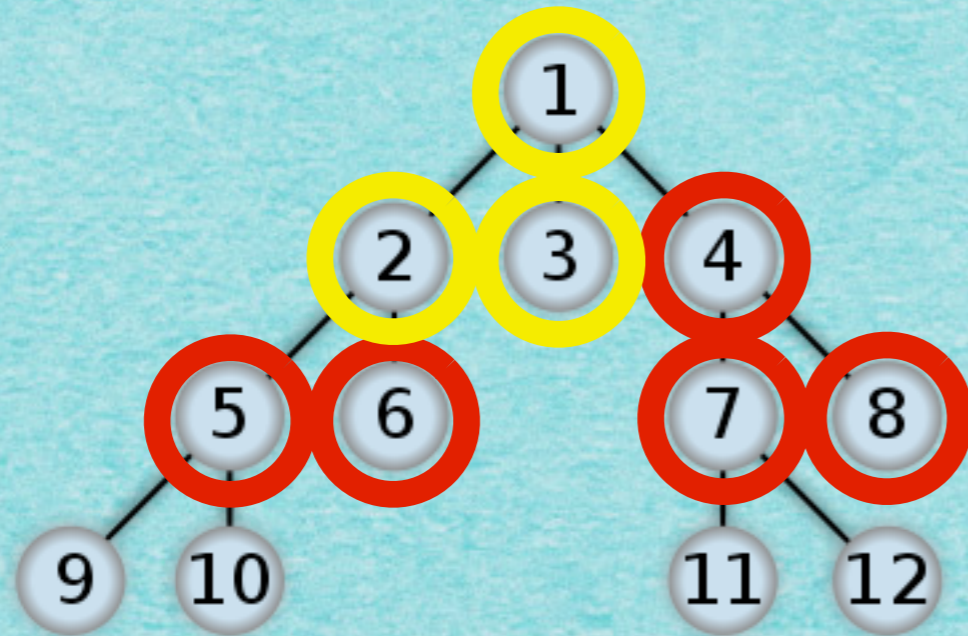
R: 4,5,6

# Graphenscan mit WARTESCHLANGE



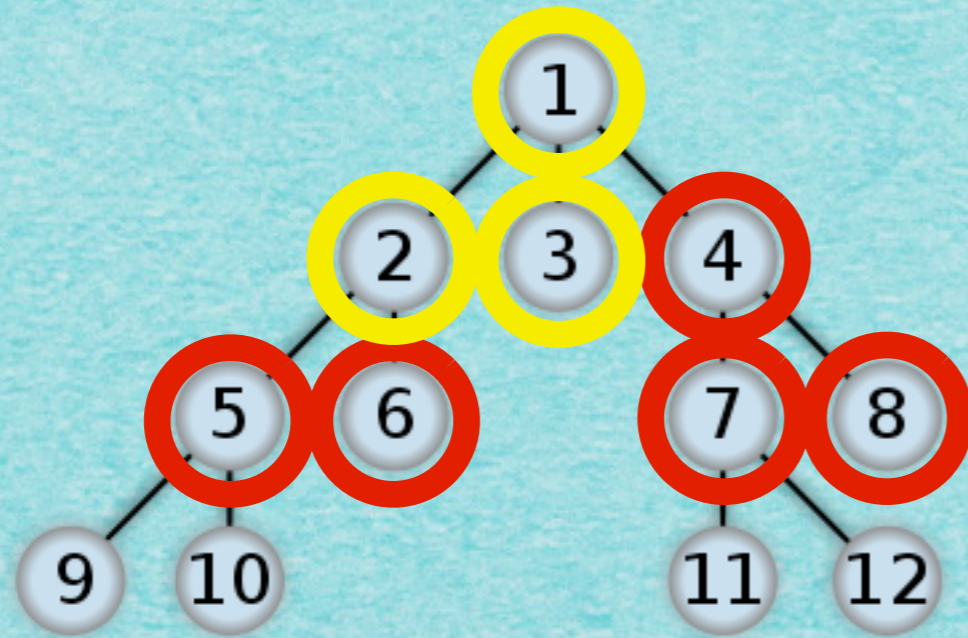
R: 4,5,6,7

# Graphenscan mit WARTESCHLANGE



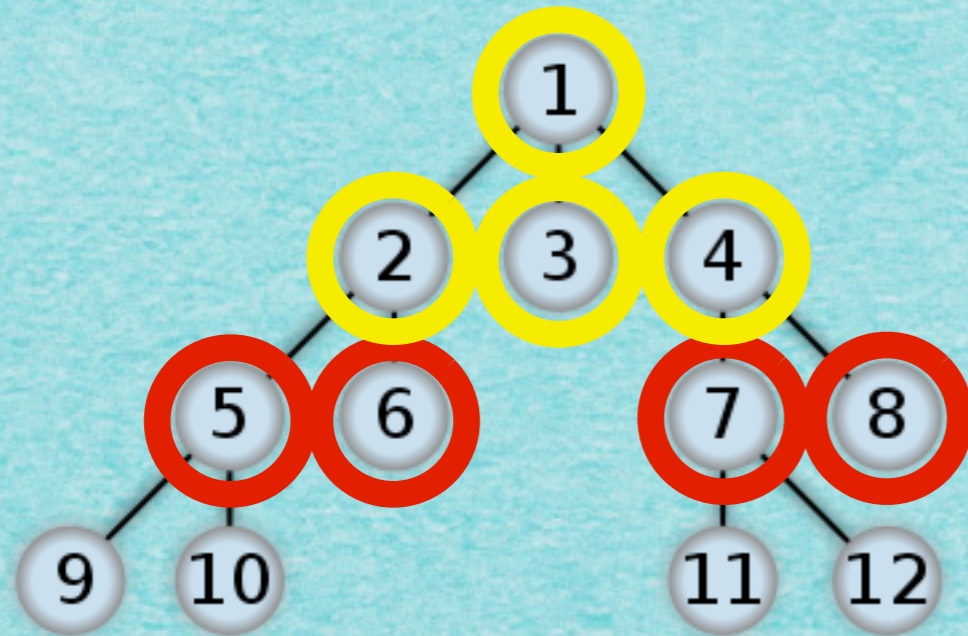
R: 4,5,6,7

# Graphenscan mit WARTESCHLANGE



R: 4,5,6,7,8

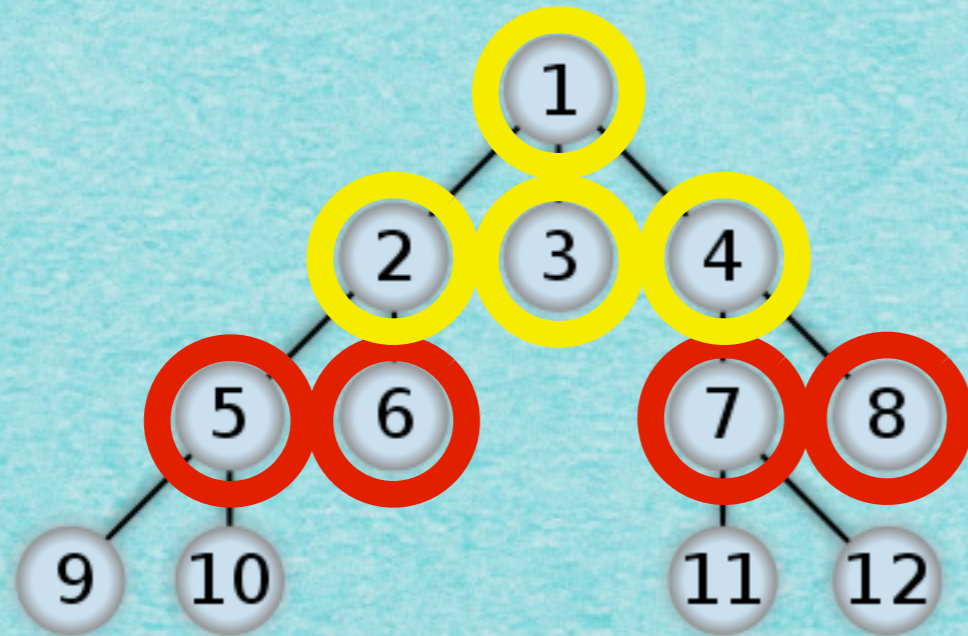
# Graphenscan mit WARTESCHLANGE



R: 4,5,6,7,8

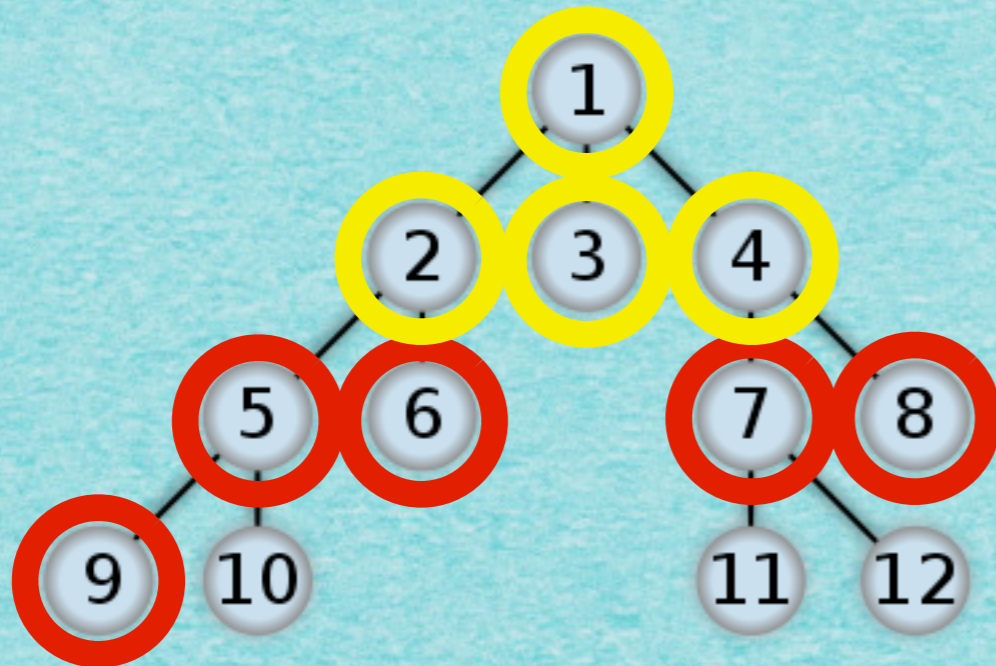


# Graphenscan mit WARTESCHLANGE



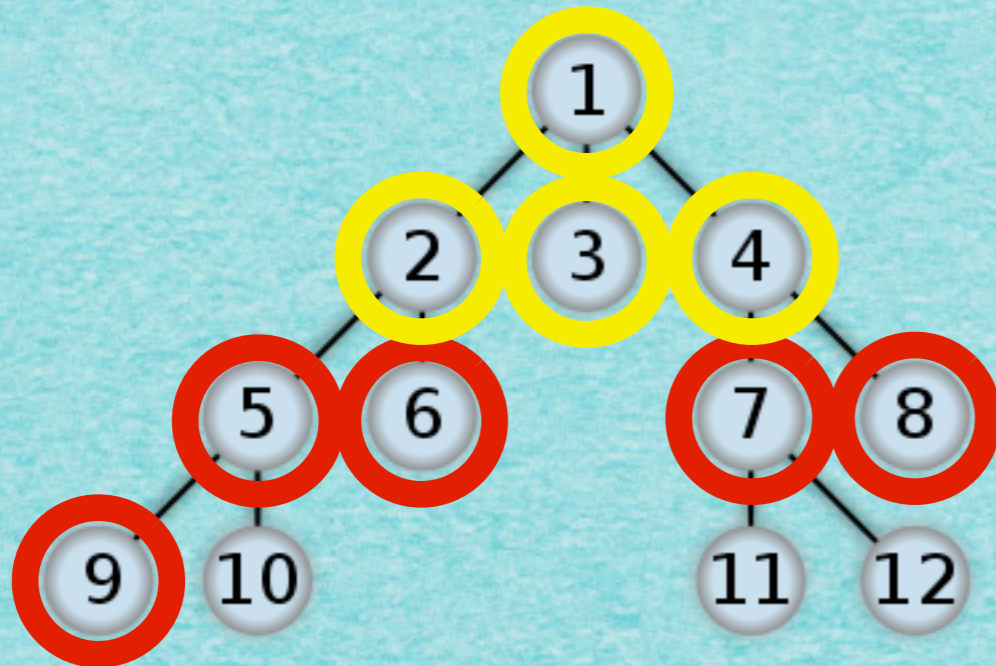
R: 5,6,7,8

# Graphenscan mit WARTESCHLANGE



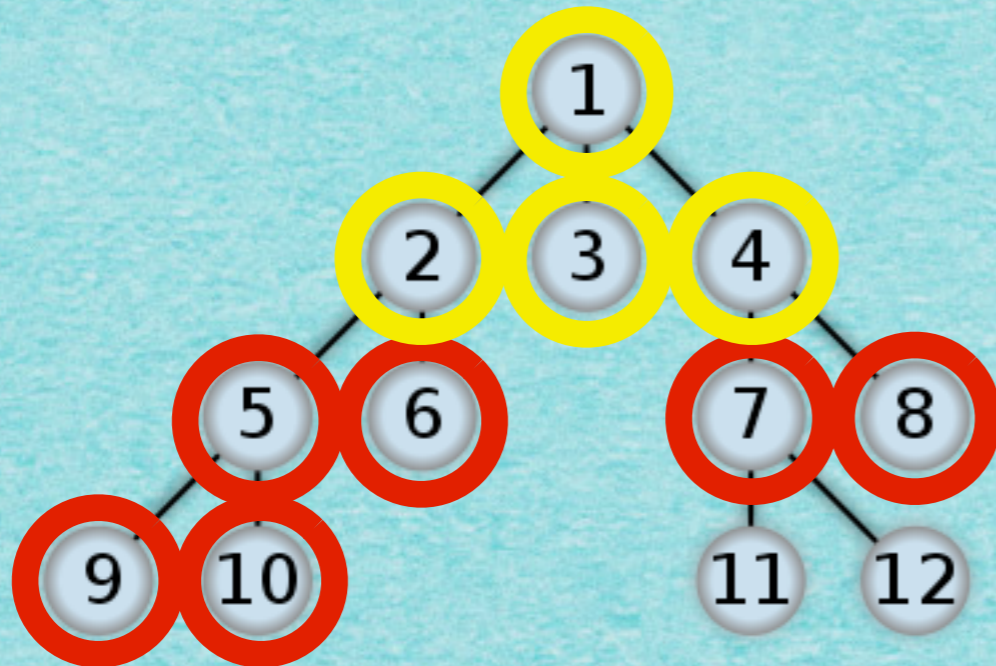
R: 5,6,7,8

# Graphenscan mit WARTESCHLANGE



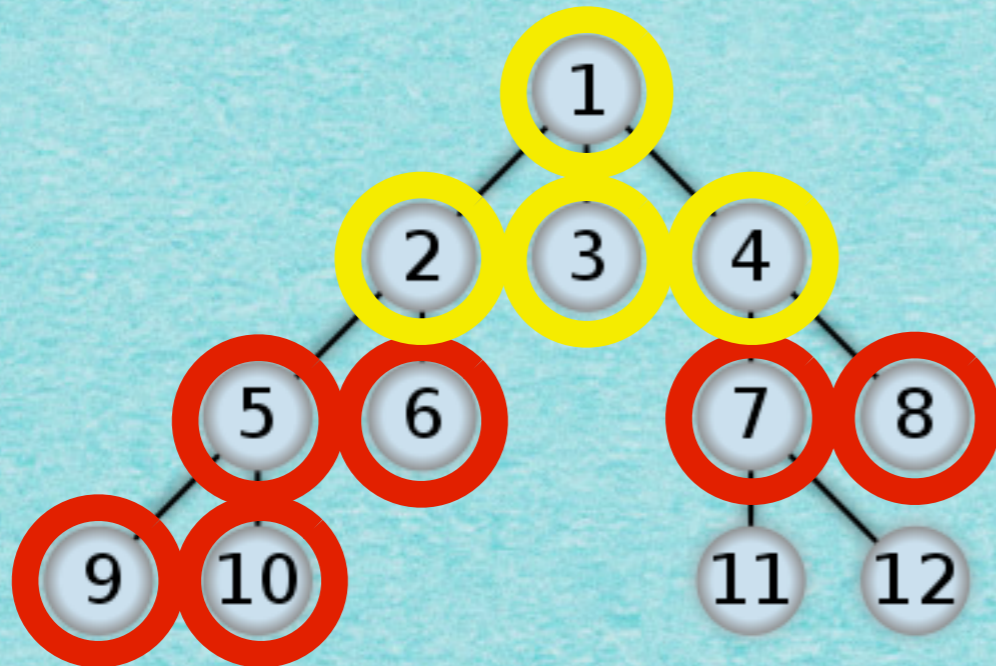
R: 5,6,7,8,9

# Graphenscan mit WARTESCHLANGE



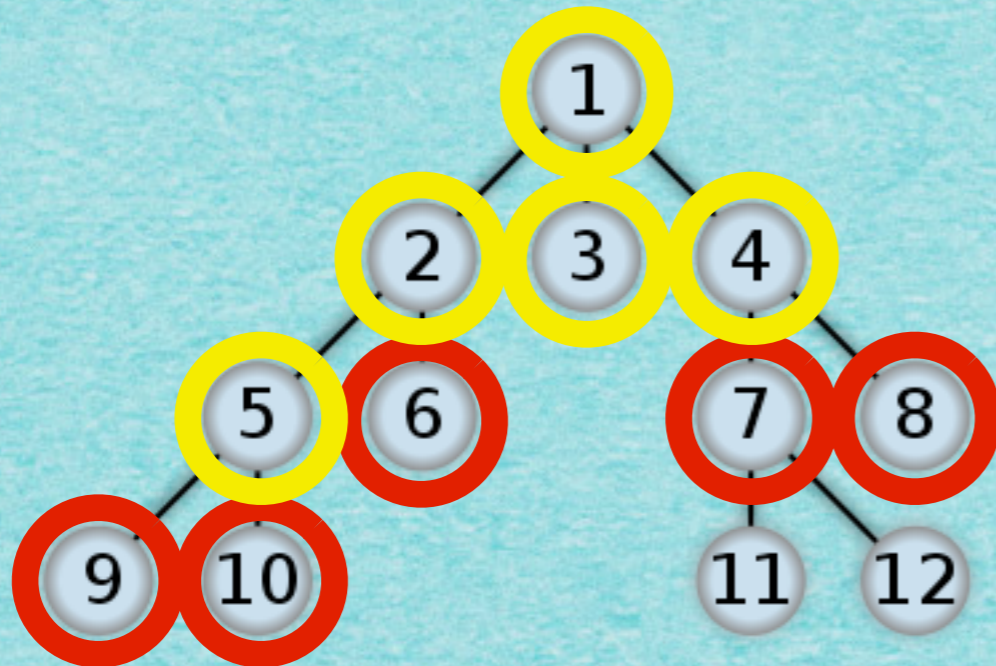
R: 5,6,7,8,9

# Graphenscan mit WARTESCHLANGE



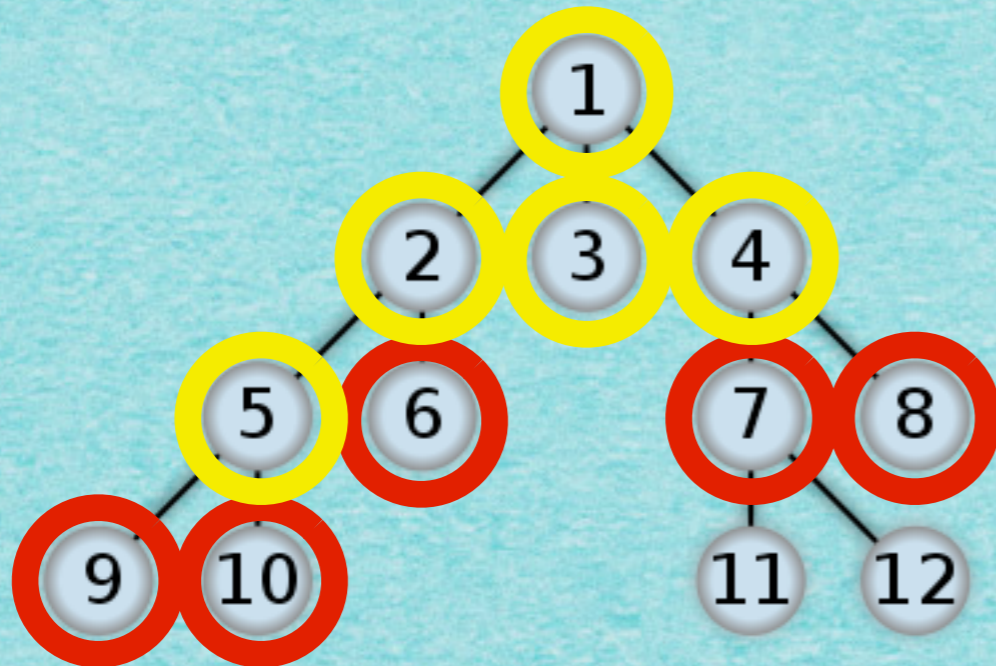
R: 5,6,7,8,9,10

# Graphenscan mit WARTESCHLANGE



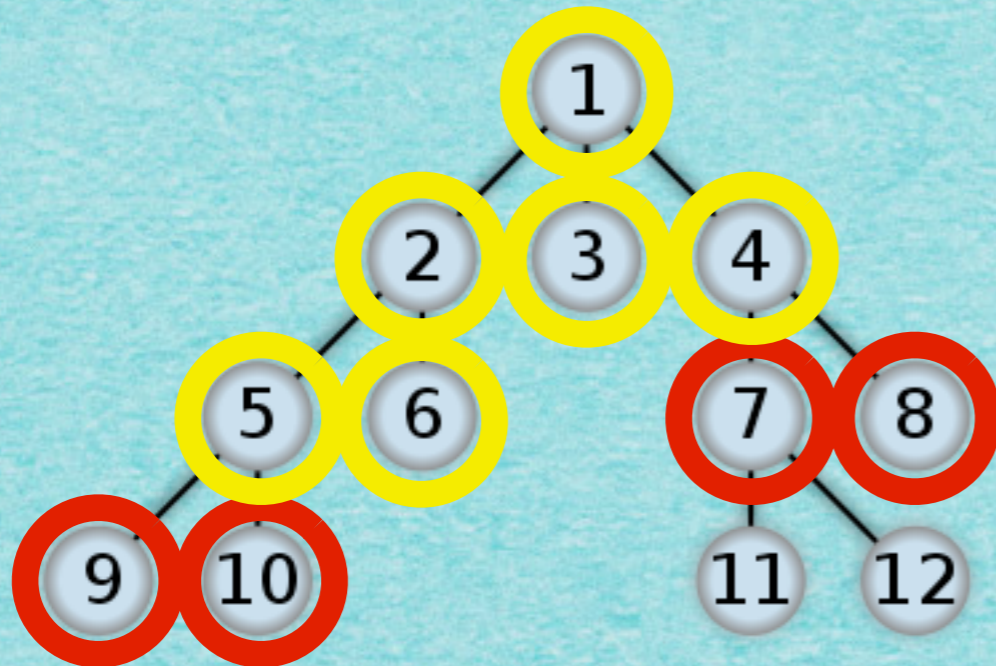
R: 5,6,7,8,9,10

# Graphenscan mit WARTESCHLANGE



R: 6,7,8,9,10

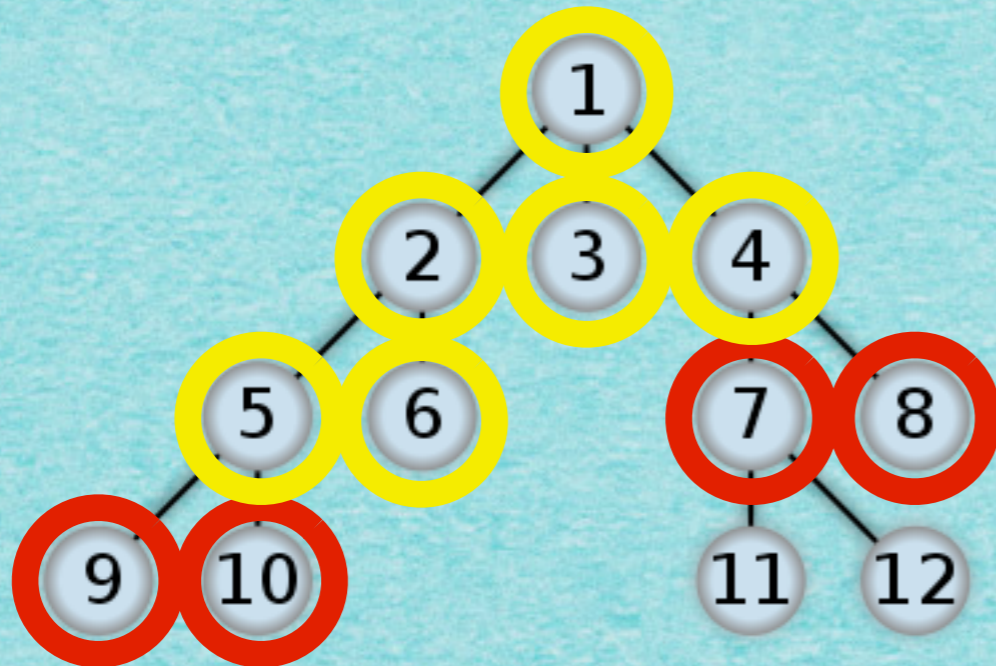
# Graphenscan mit WARTESCHLANGE



R: 6,7,8,9,10

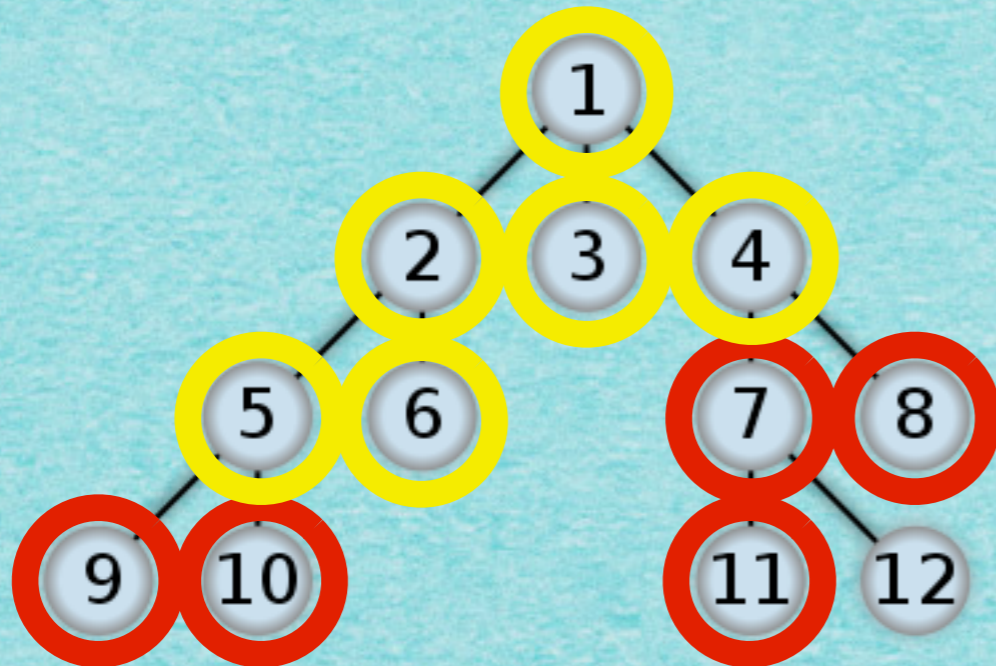


# Graphenscan mit WARTESCHLANGE



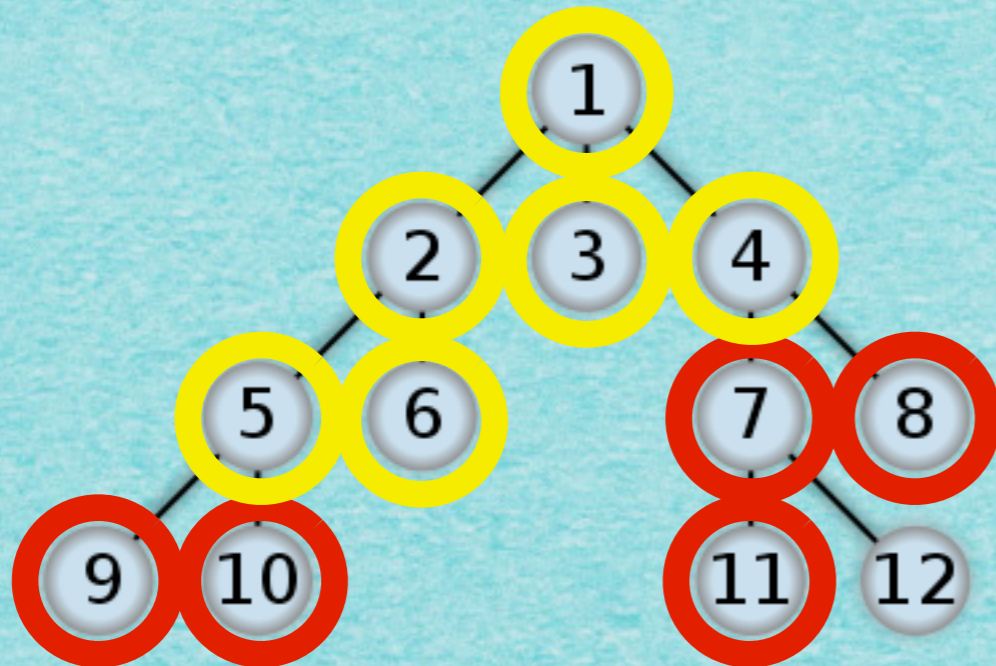
R: 7,8,9,10

# Graphenscan mit WARTESCHLANGE



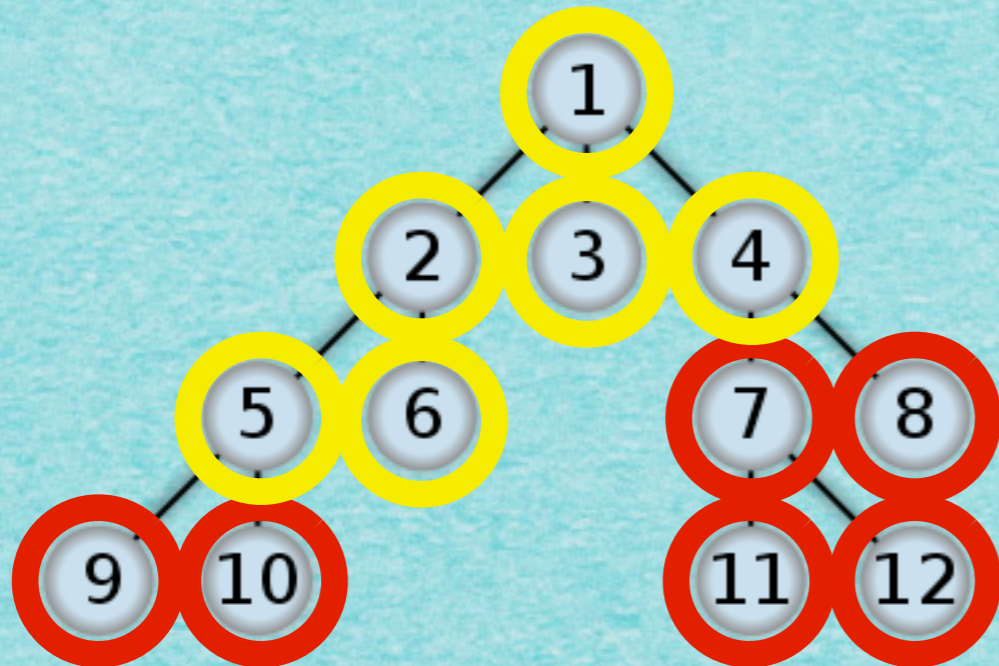
R: 7,8,9,10

# Graphenscan mit WARTESCHLANGE



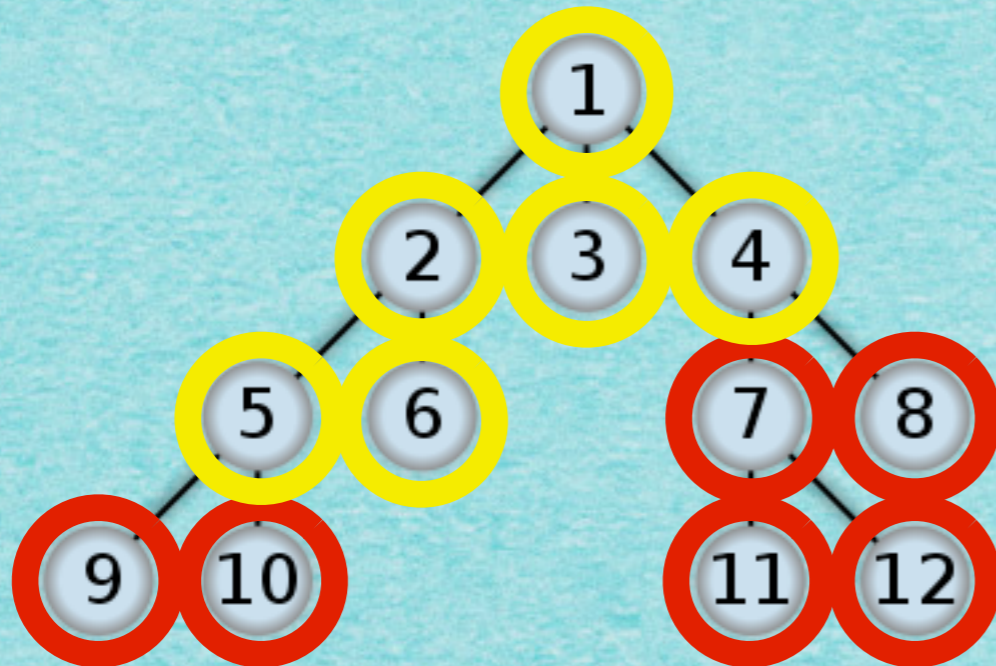
R: 7,8,9,10,11

# Graphenscan mit WARTESCHLANGE



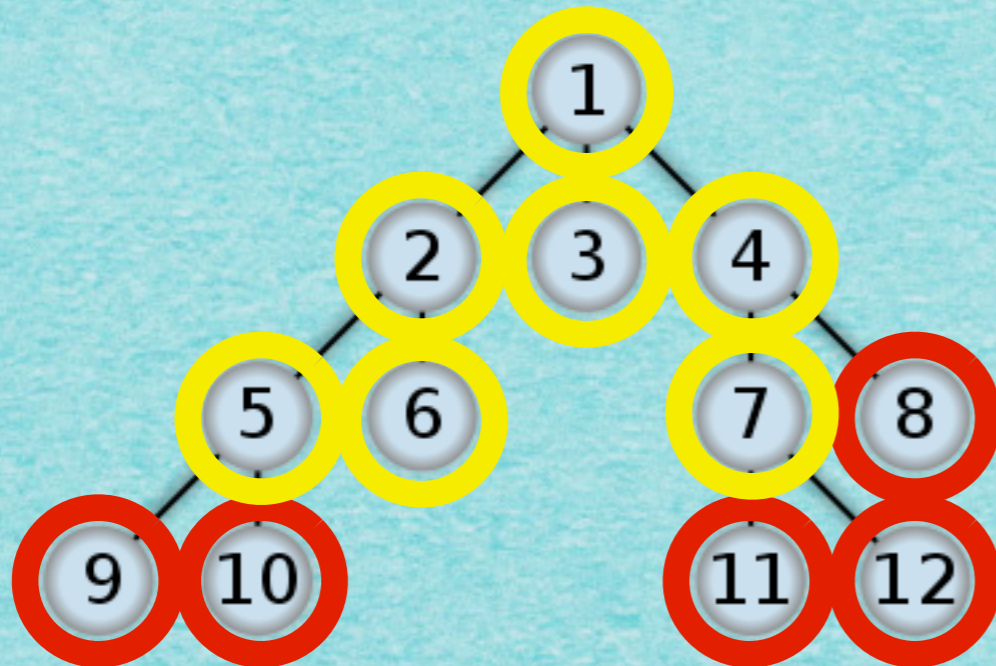
R: 7,8,9,10,11

# Graphenscan mit WARTESCHLANGE



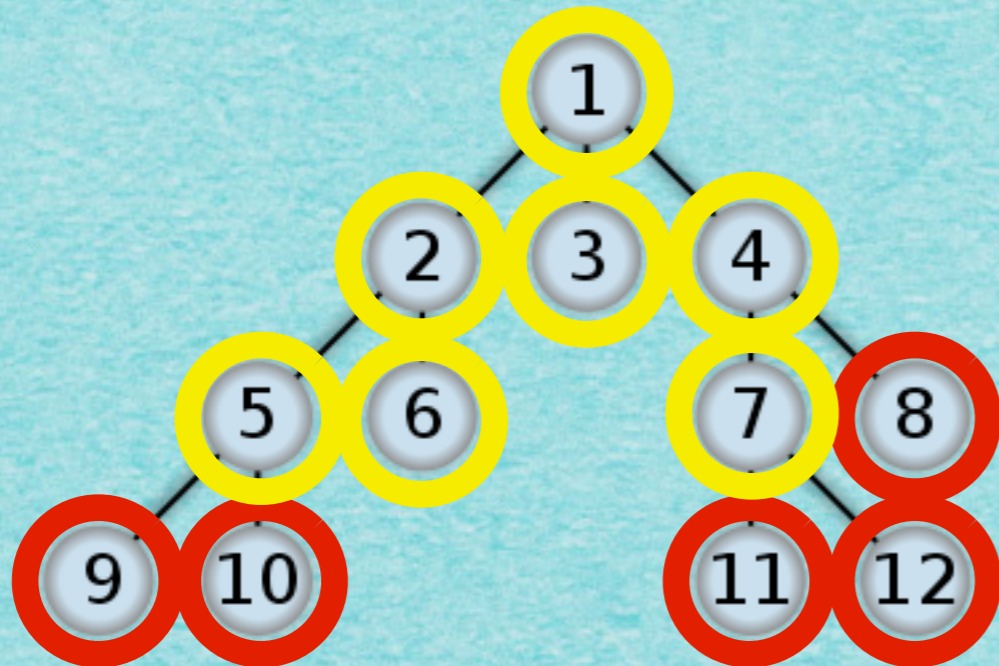
R: 7,8,9,10,11,12

# Graphenscan mit WARTESCHLANGE



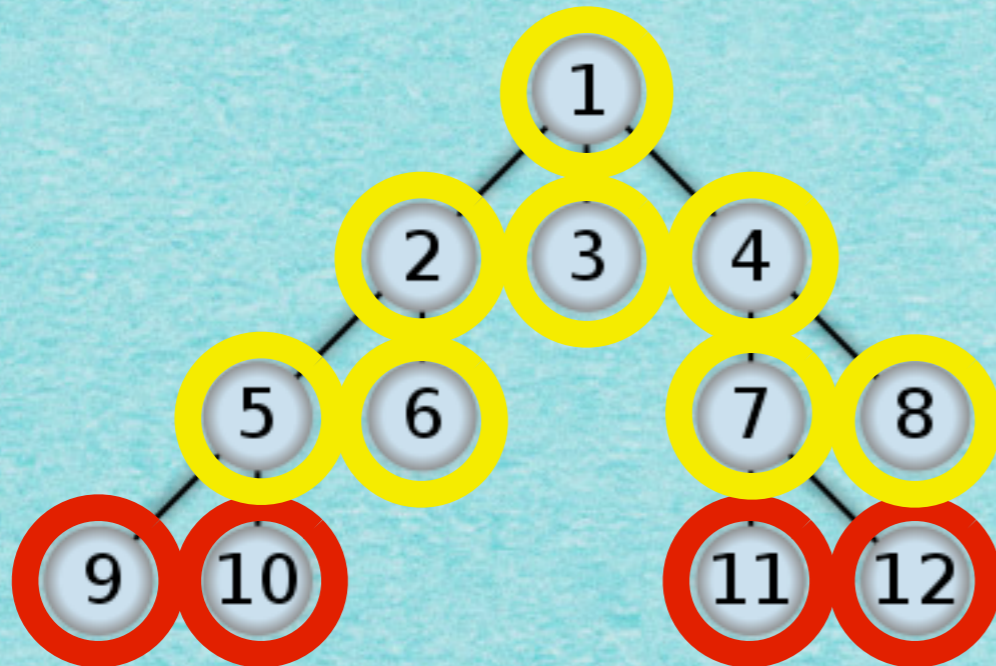
R: 7,8,9,10,11,12

# Graphenscan mit WARTESCHLANGE



R: 8,9,10,11,12

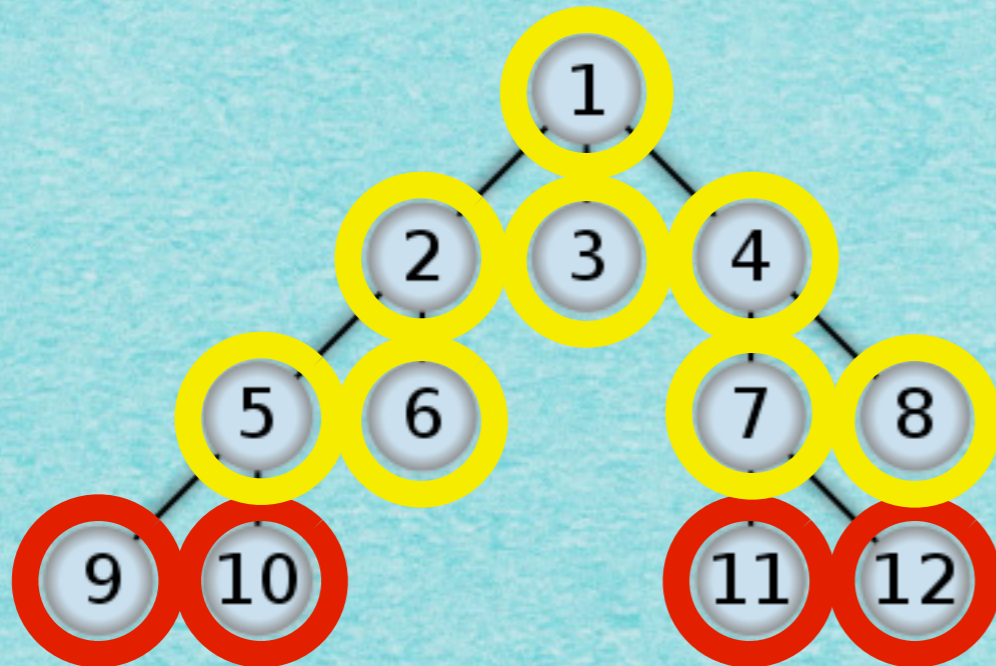
# Graphenscan mit WARTESCHLANGE



R: 8,9,10,11,12

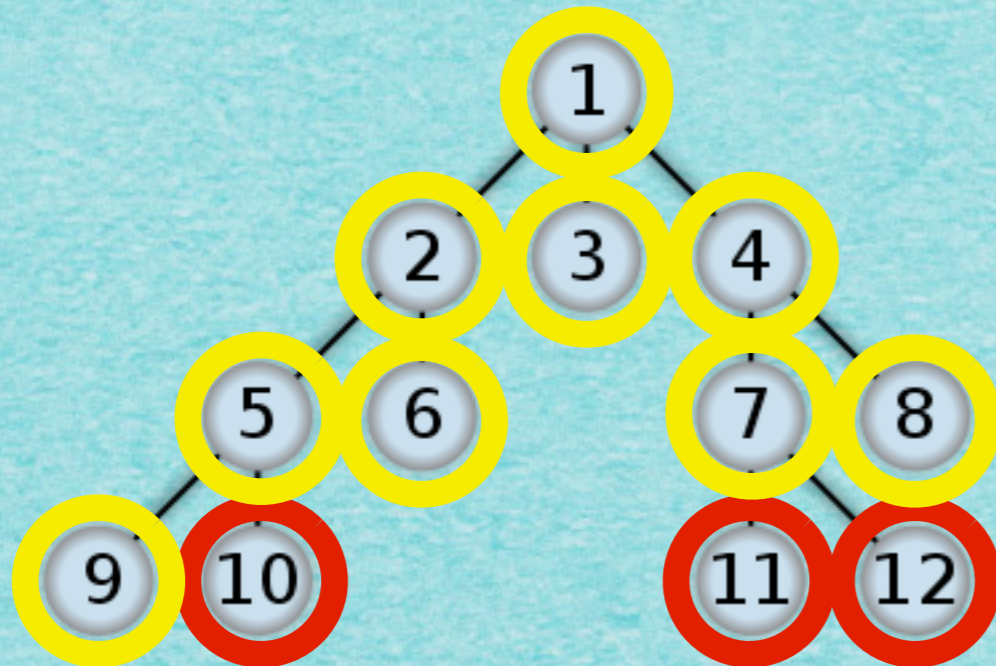


# Graphenscan mit WARTESCHLANGE



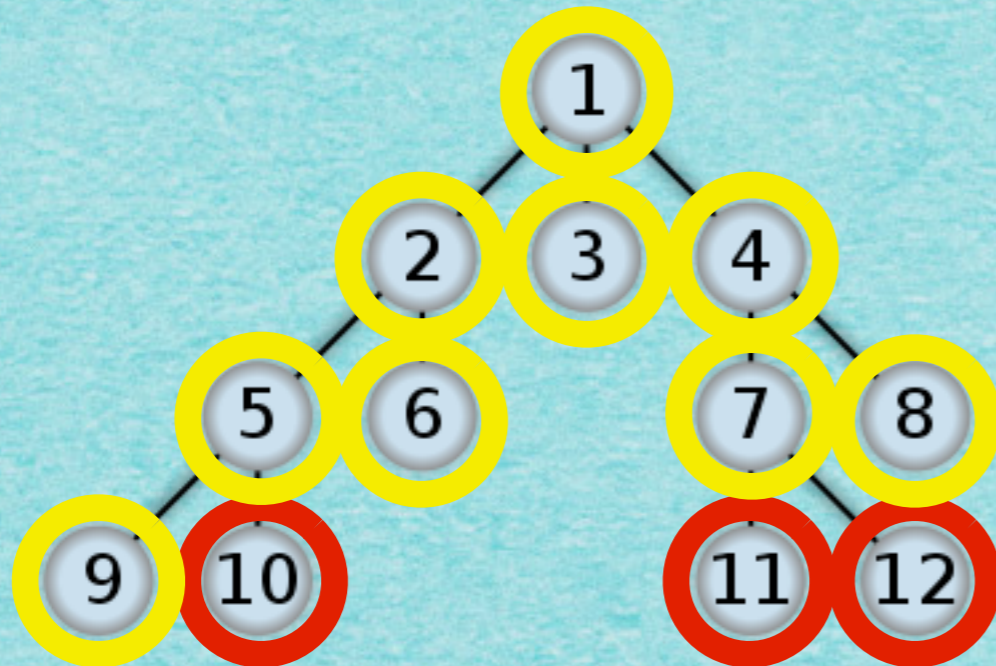
R: 9,10,11,12

# Graphenscan mit WARTESCHLANGE



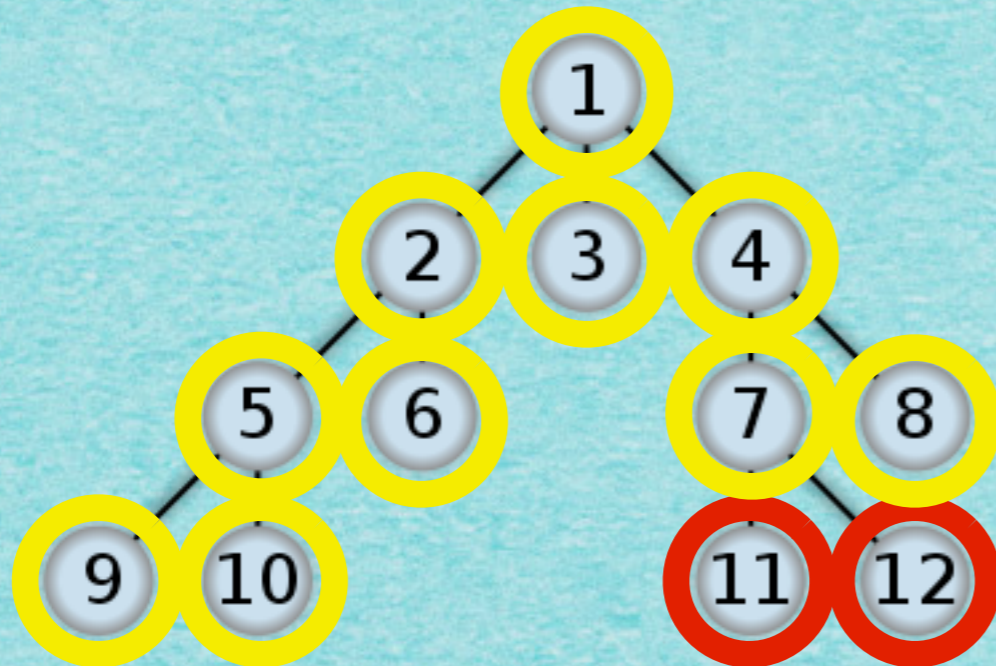
R: 9,10,11,12

# Graphenscan mit WARTESCHLANGE



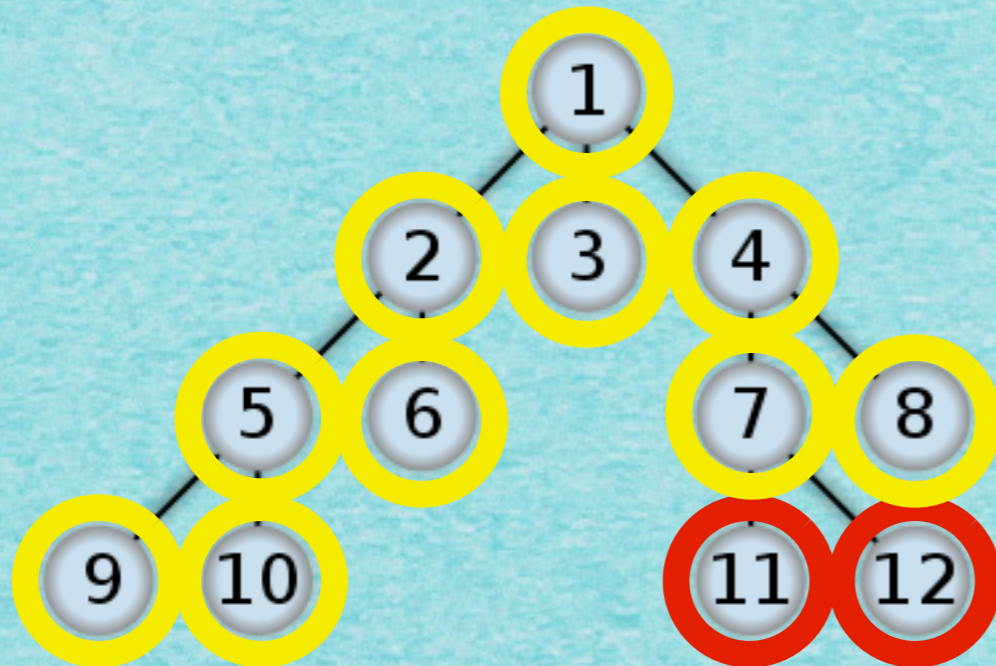
R:10,11,12

# Graphenscan mit WARTESCHLANGE



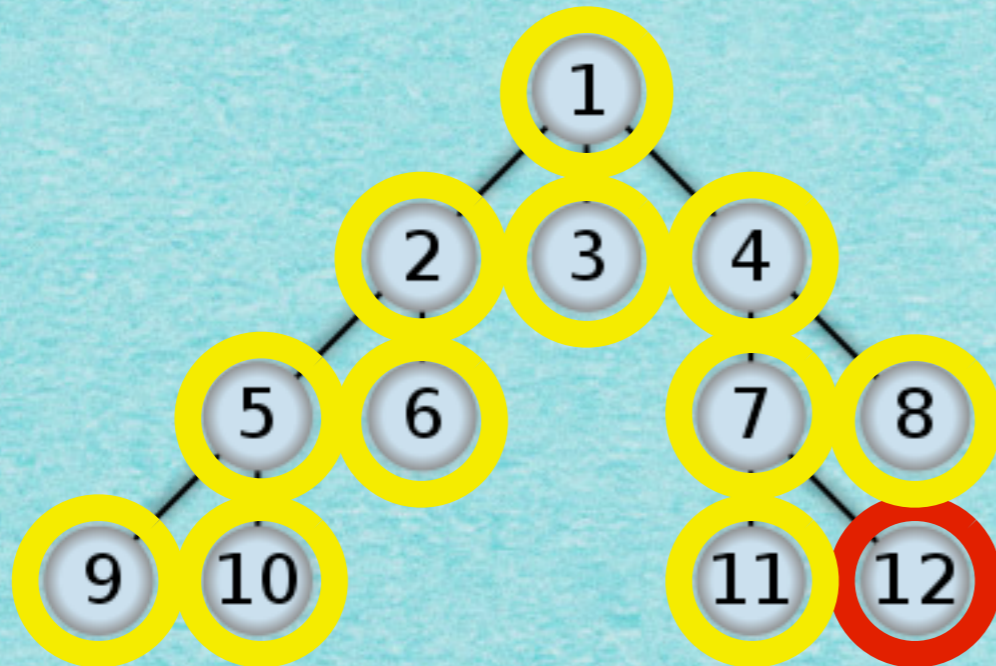
R:10,11,12

# Graphenscan mit WARTESCHLANGE



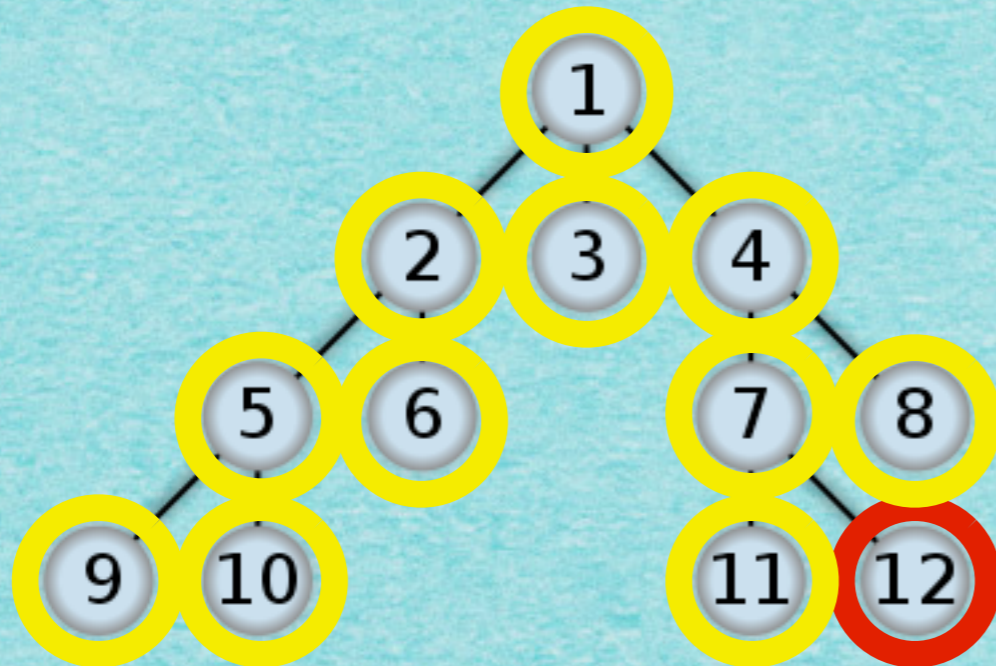
R:1 1, 12

# Graphenscan mit WARTESCHLANGE



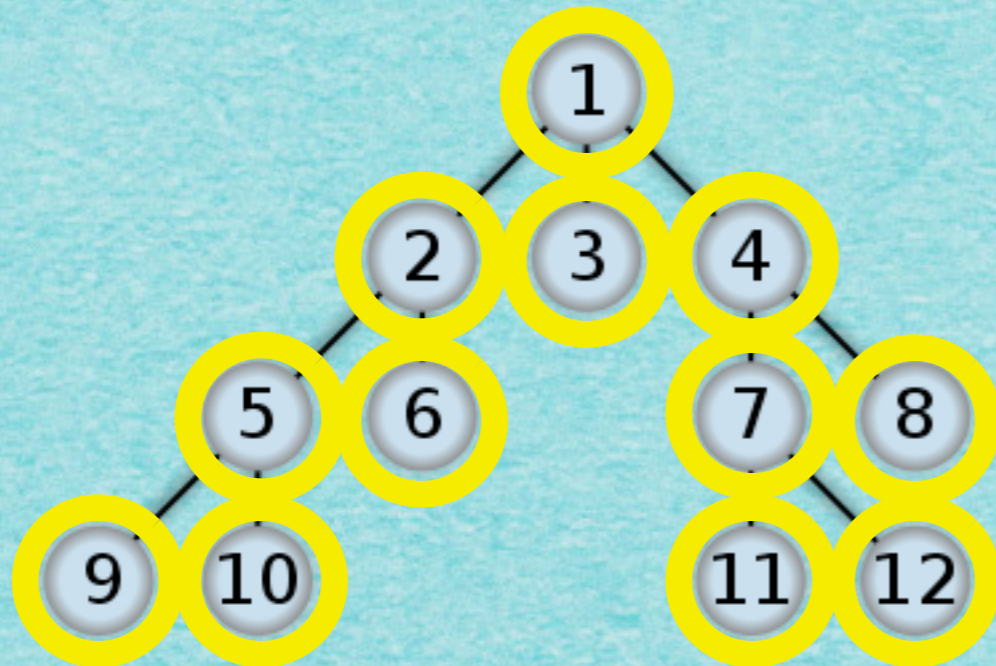
R:1 1, 12

# Graphenscan mit WARTESCHLANGE



R:12

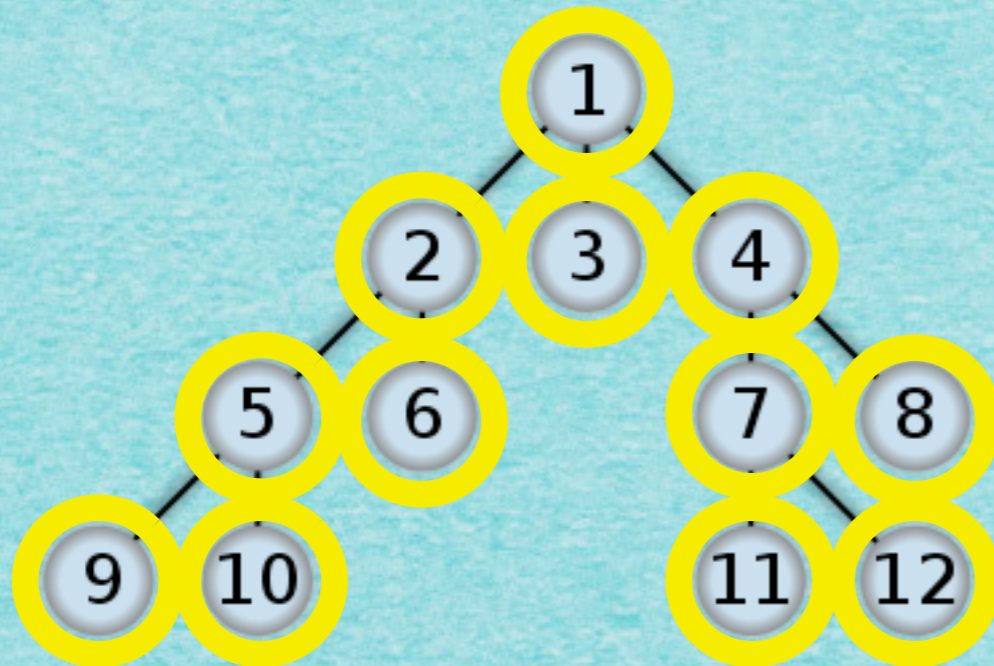
# Graphenscan mit WARTESCHLANGE



R:12

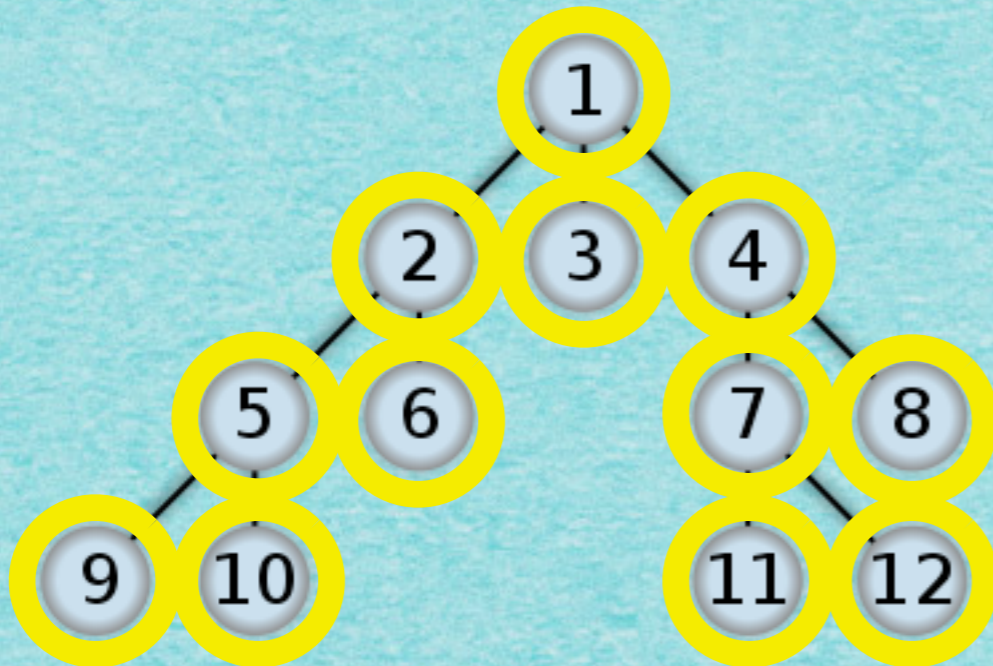


# Graphenscan mit WARTESCHLANGE



R:

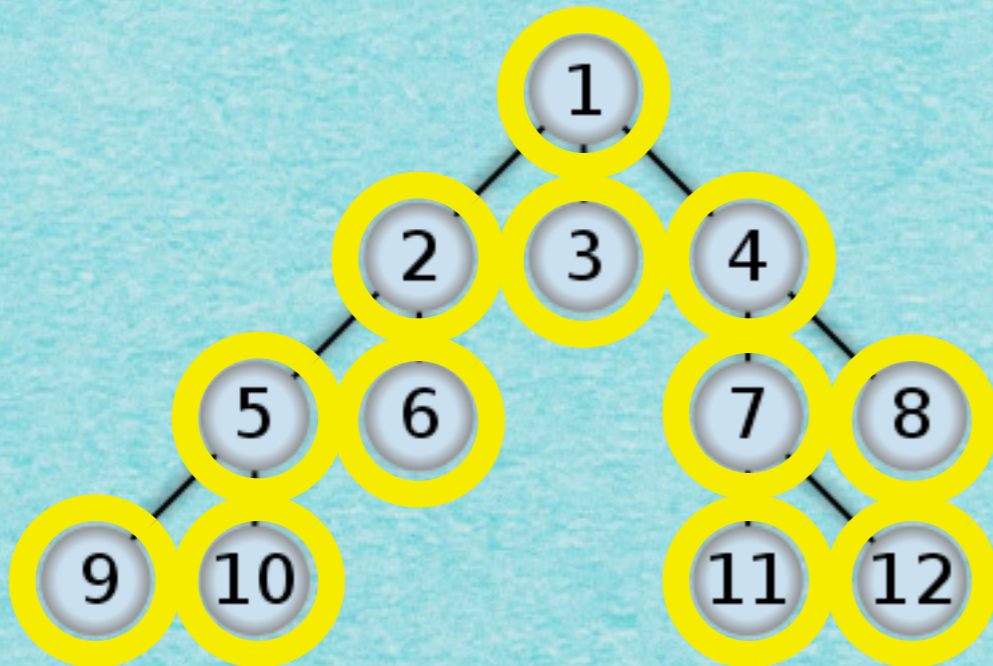
# Graphenscan mit WARTESCHLANGE



R:

**STOP!**

# Graphenscan mit WARTESCHLANGE

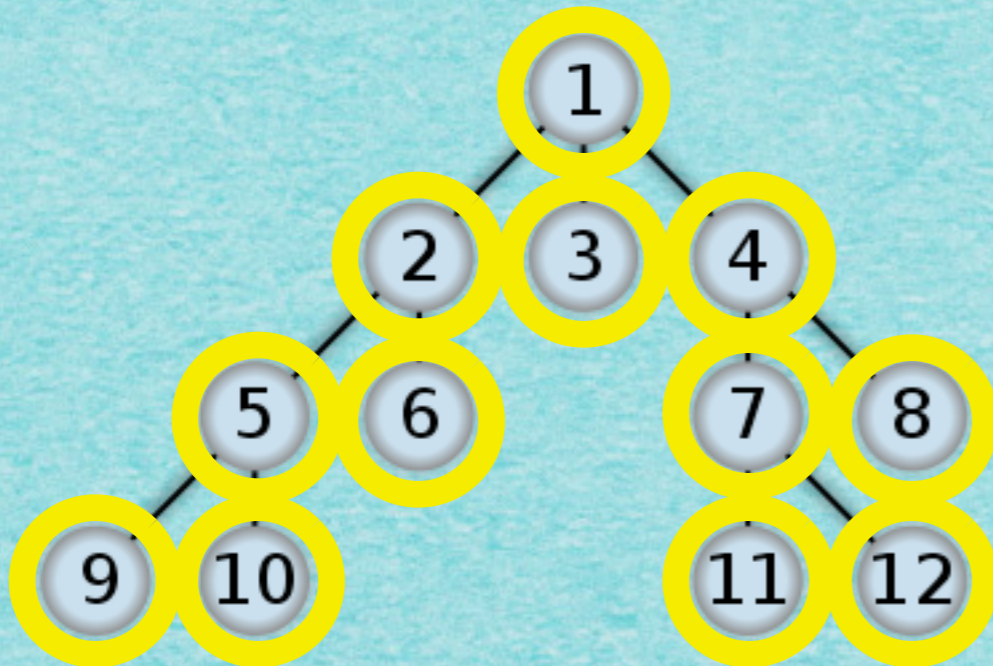


R:

**STOP!**

**BREITENSUCHE - "BREADTH-FIRST SEARCH" (BFS)**

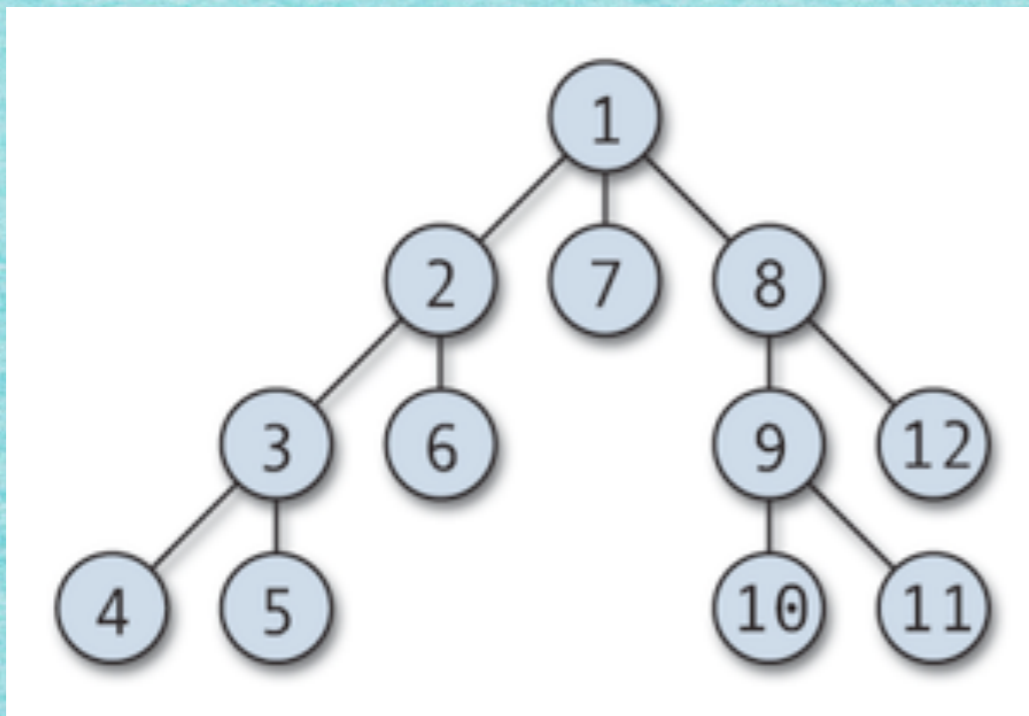
# Graphenscan mit WARTESCHLANGE



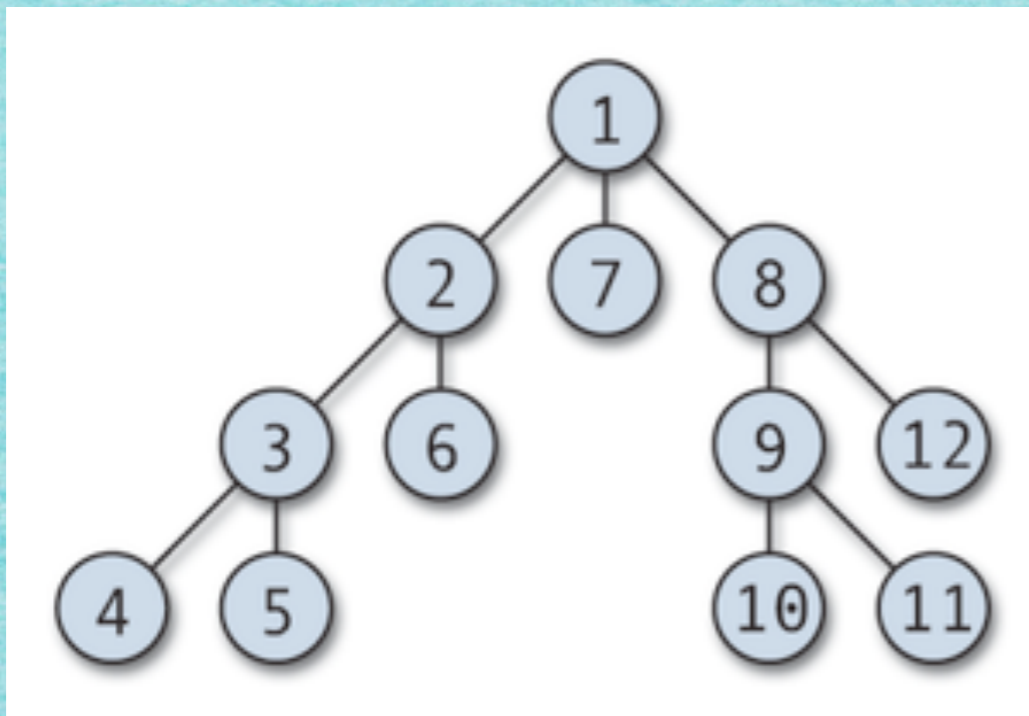
R:

**STOP!**

# Graphenscan mit STAPEL

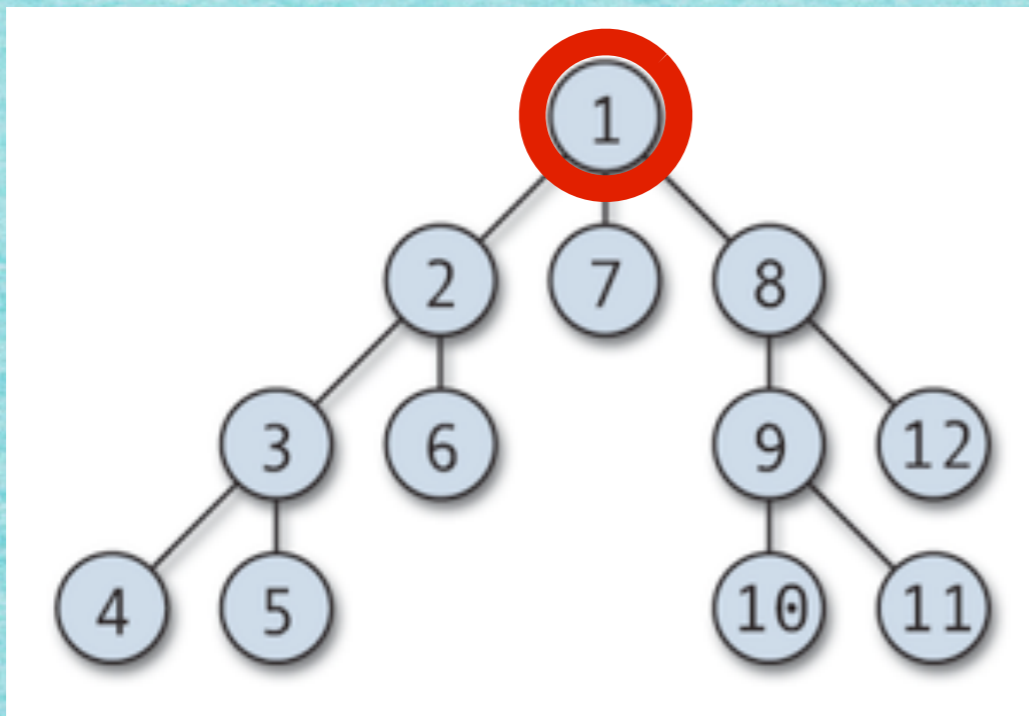


# Graphenscan mit STAPEL



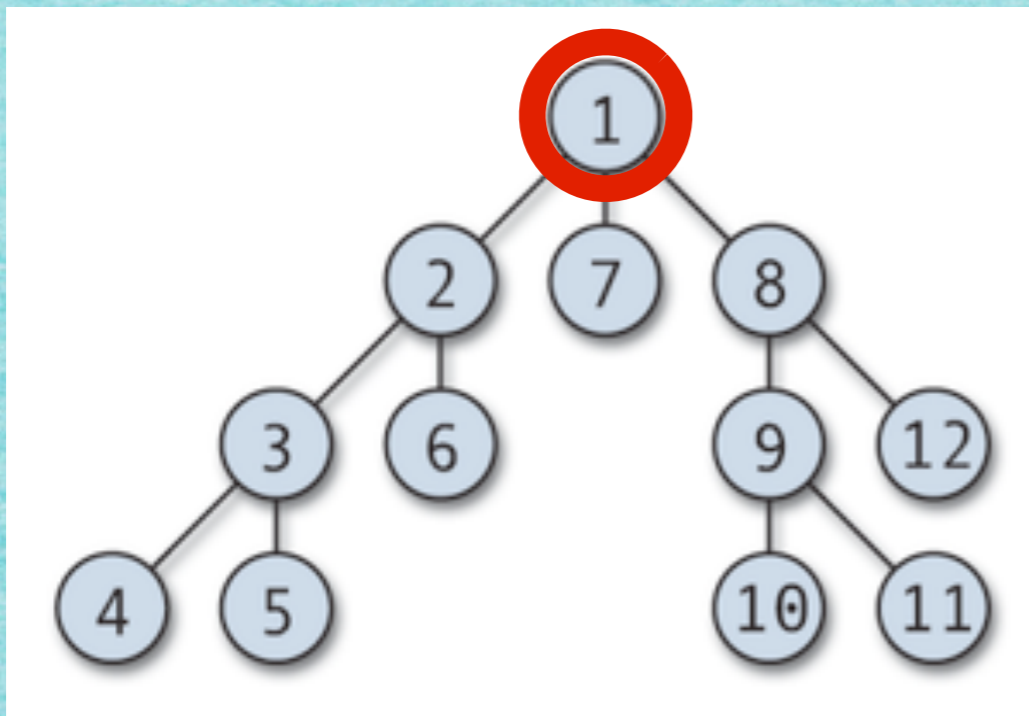
R:

# Graphenscan mit STAPEL



R:

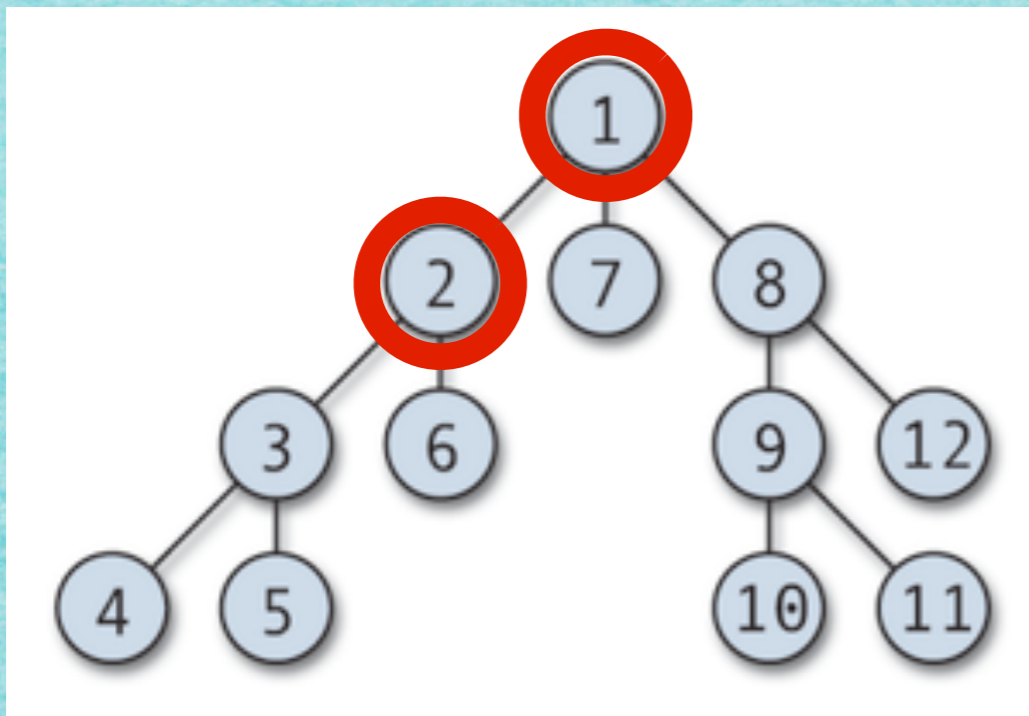
# Graphenscan mit STAPEL



R: 1

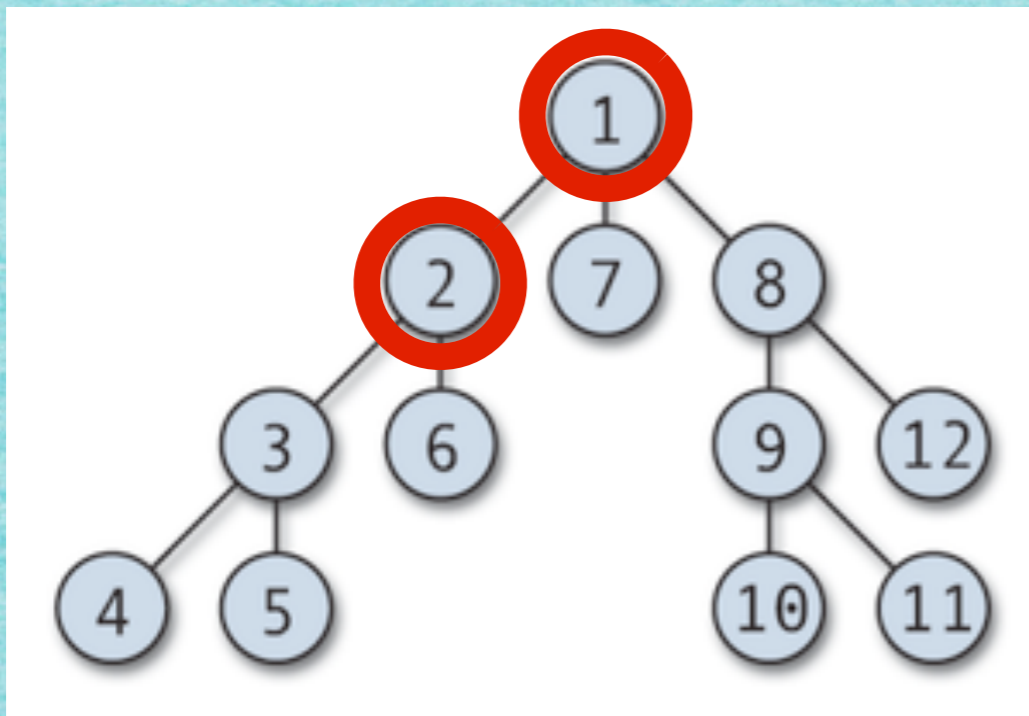


# Graphenscan mit STAPEL



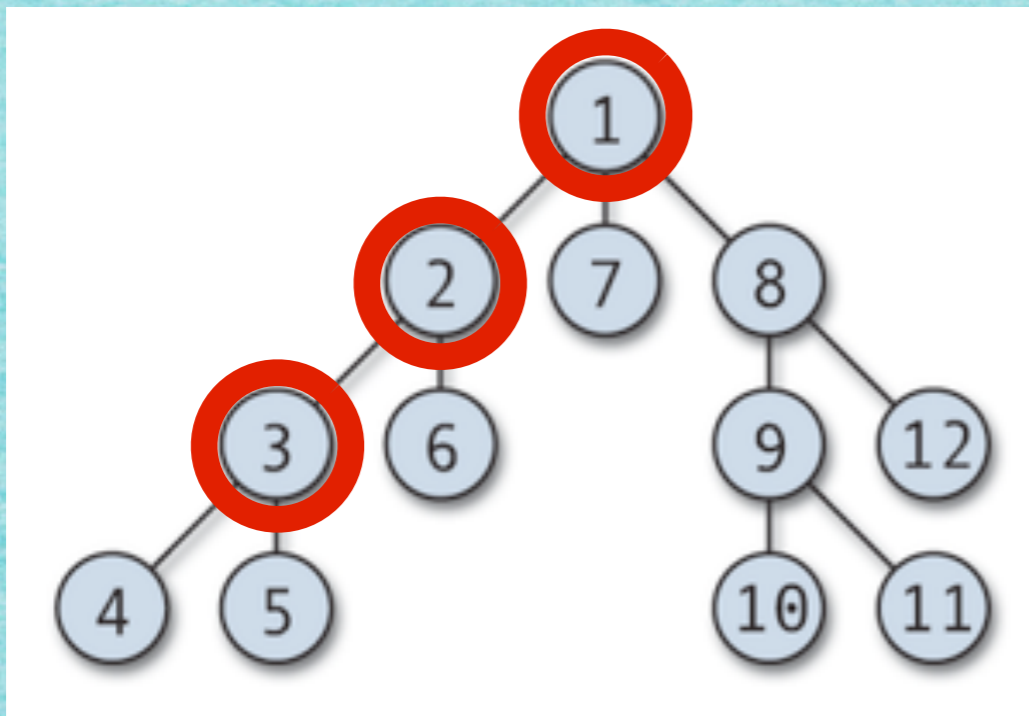
R: 1

# Graphenscan mit STAPEL



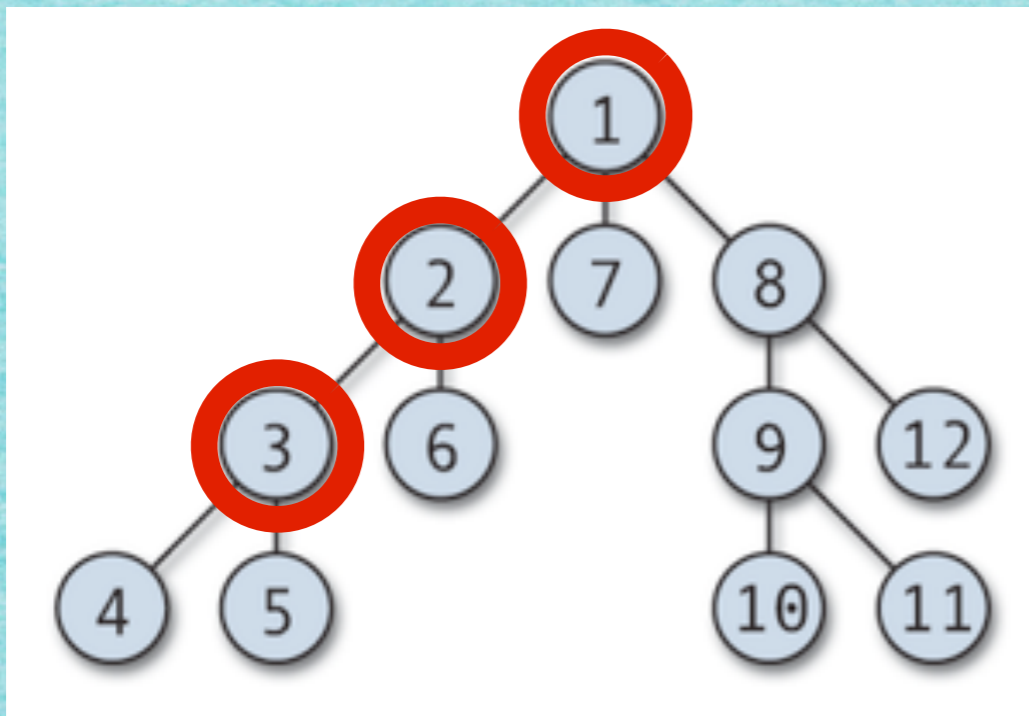
R: 1,2

# Graphenscan mit STAPEL



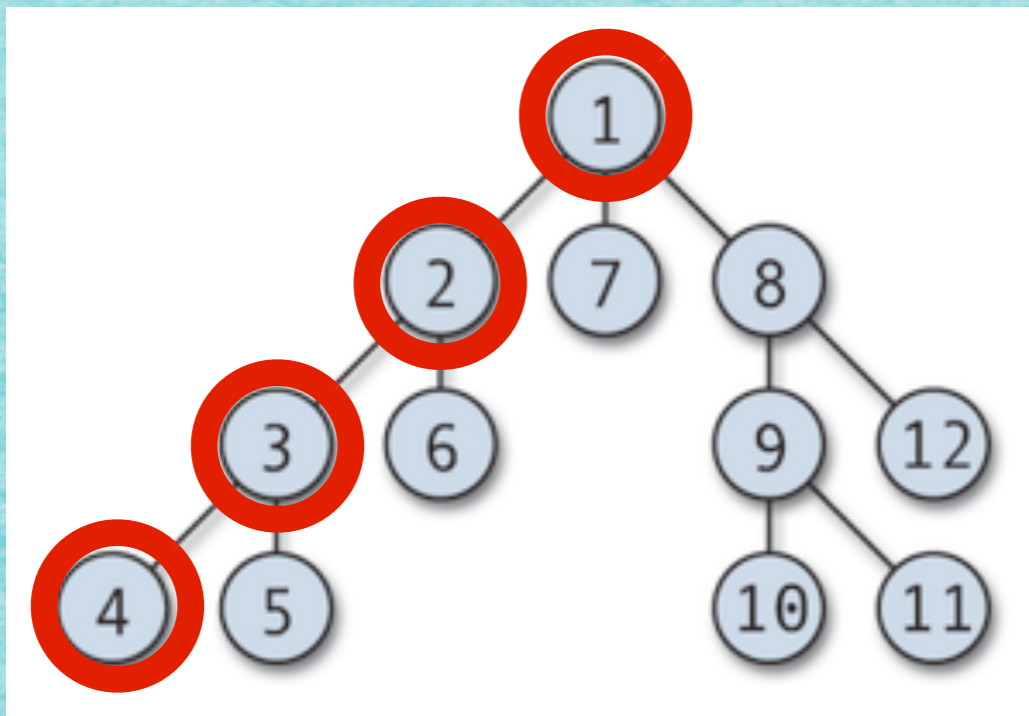
R: 1,2

# Graphenscan mit STAPEL



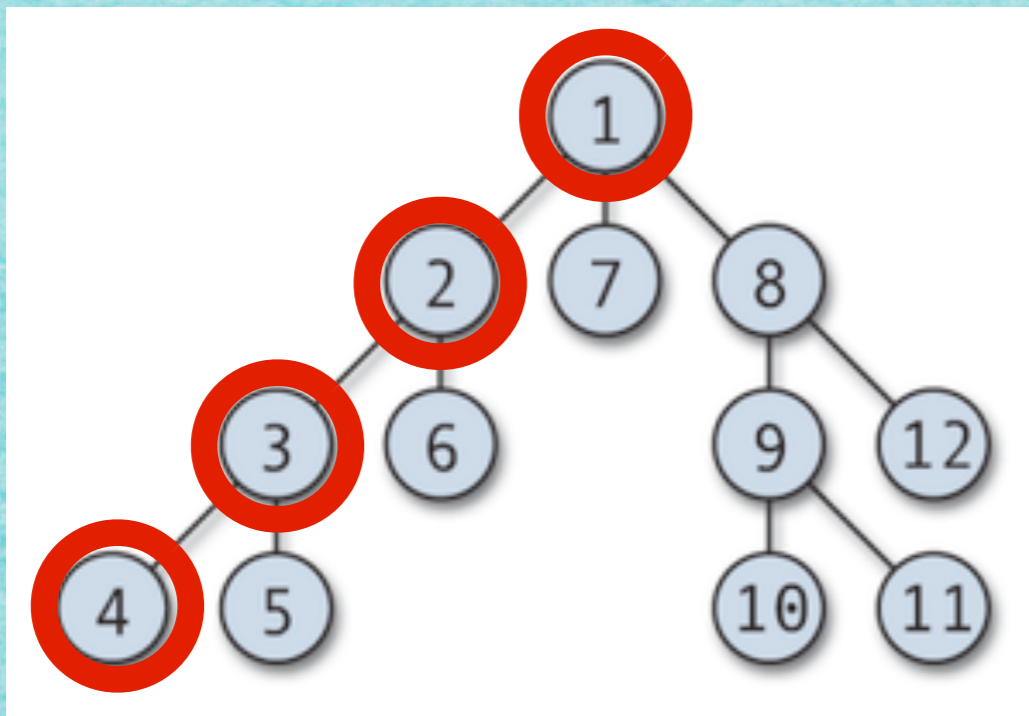
R: 1,2,3

# Graphenscan mit STAPEL



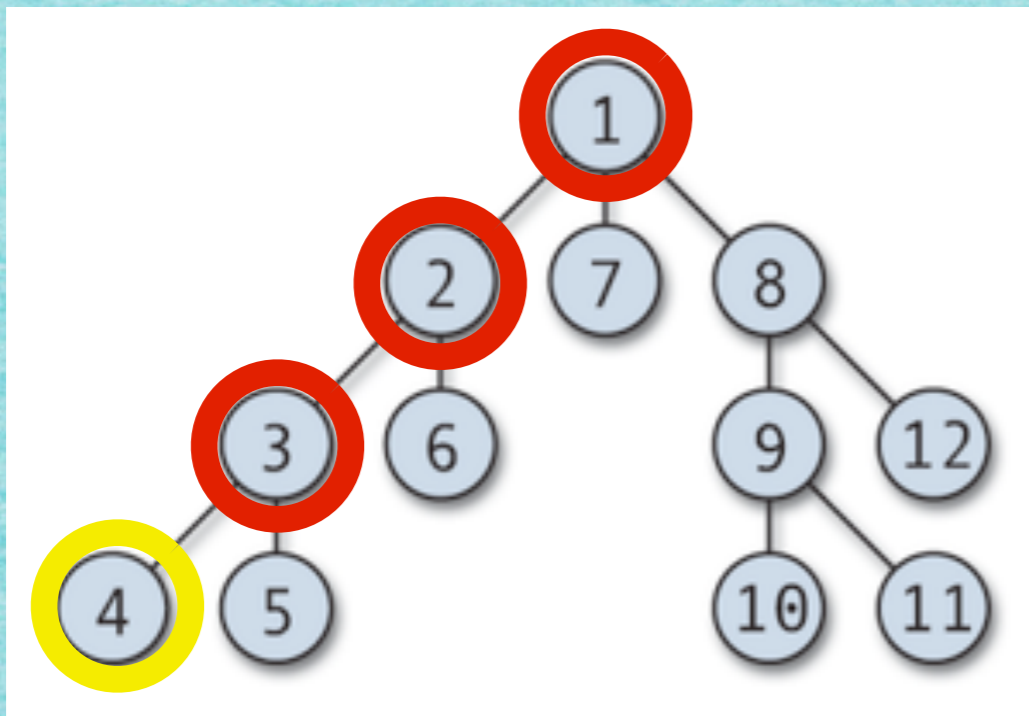
R: 1,2,3

# Graphenscan mit STAPEL



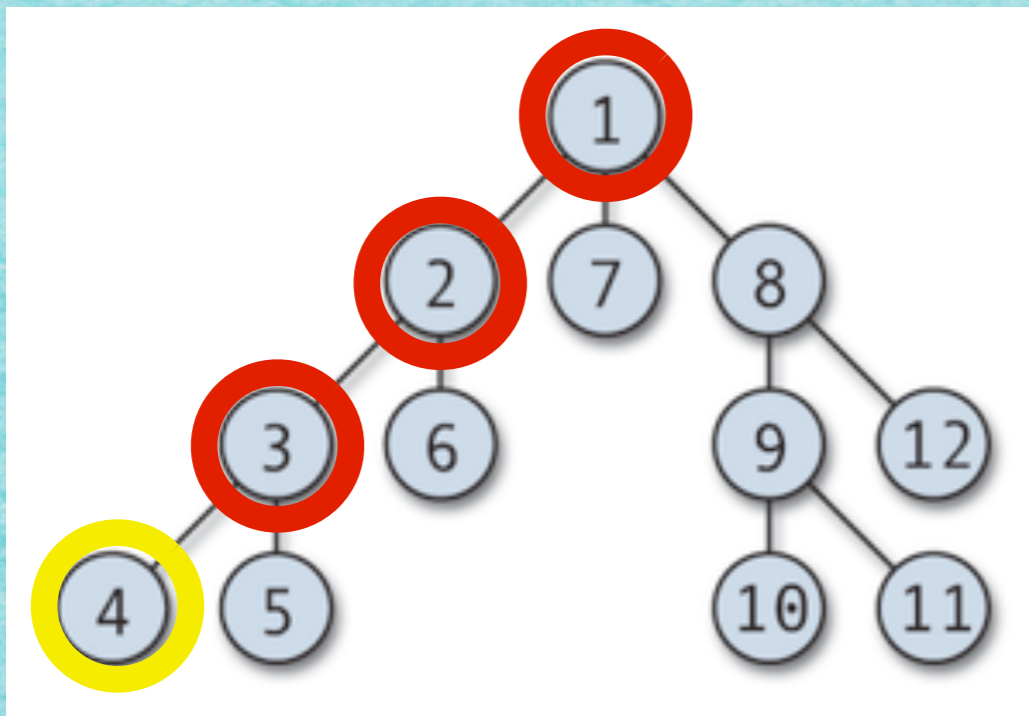
R: 1,2,3,4

# Graphenscan mit STAPEL



R: 1,2,3,4

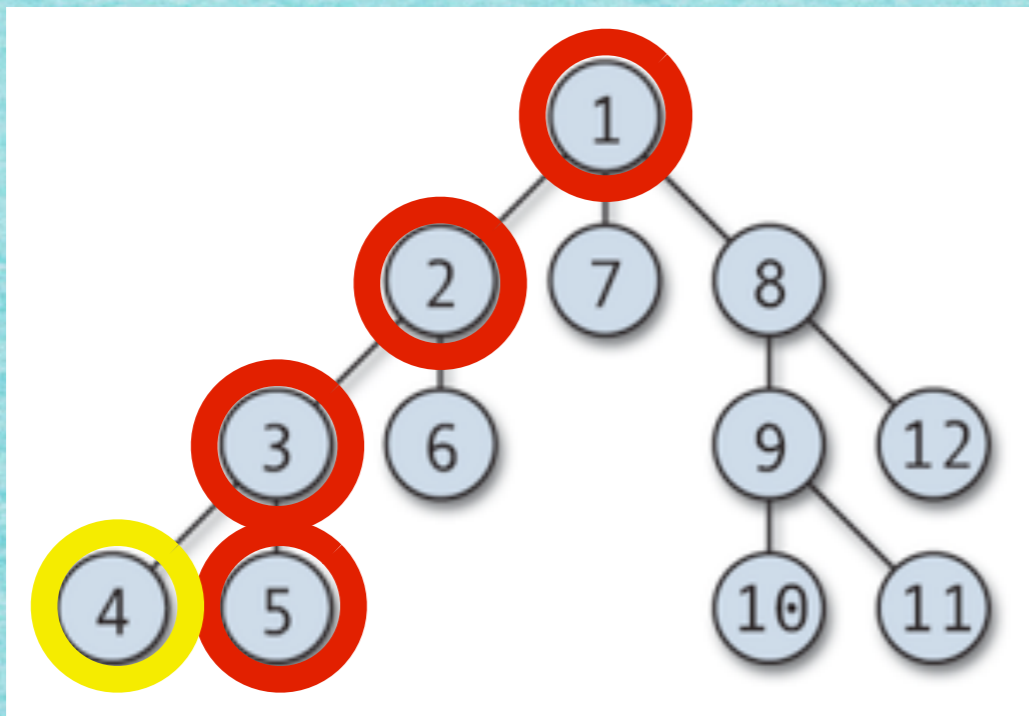
# Graphenscan mit STAPEL



R: 1,2,3

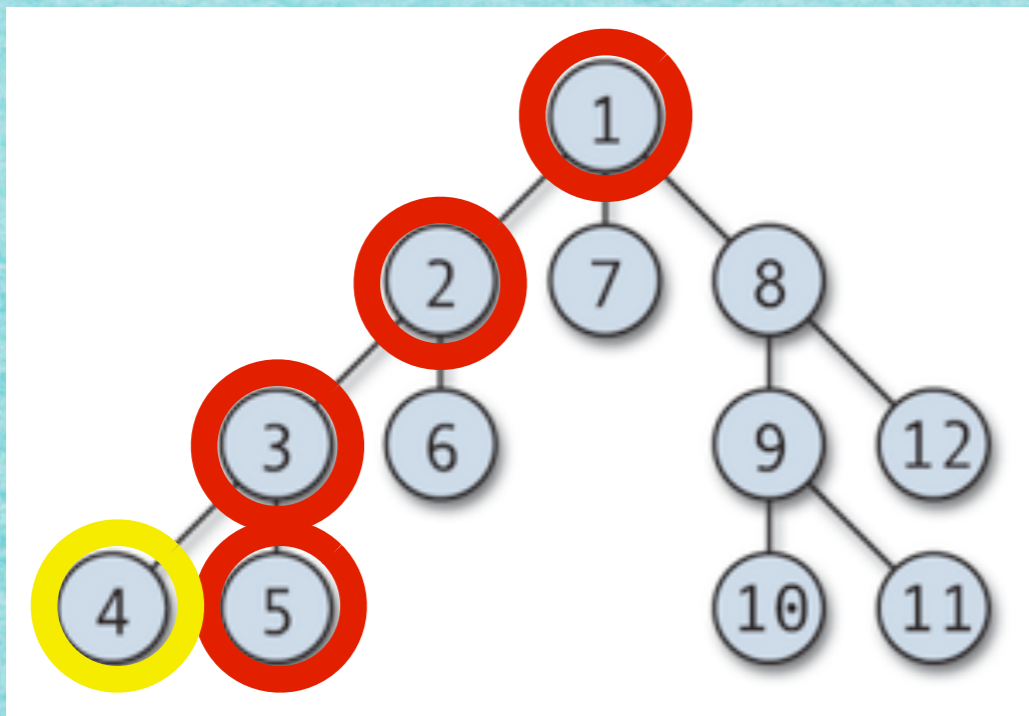


# Graphenscan mit STAPEL



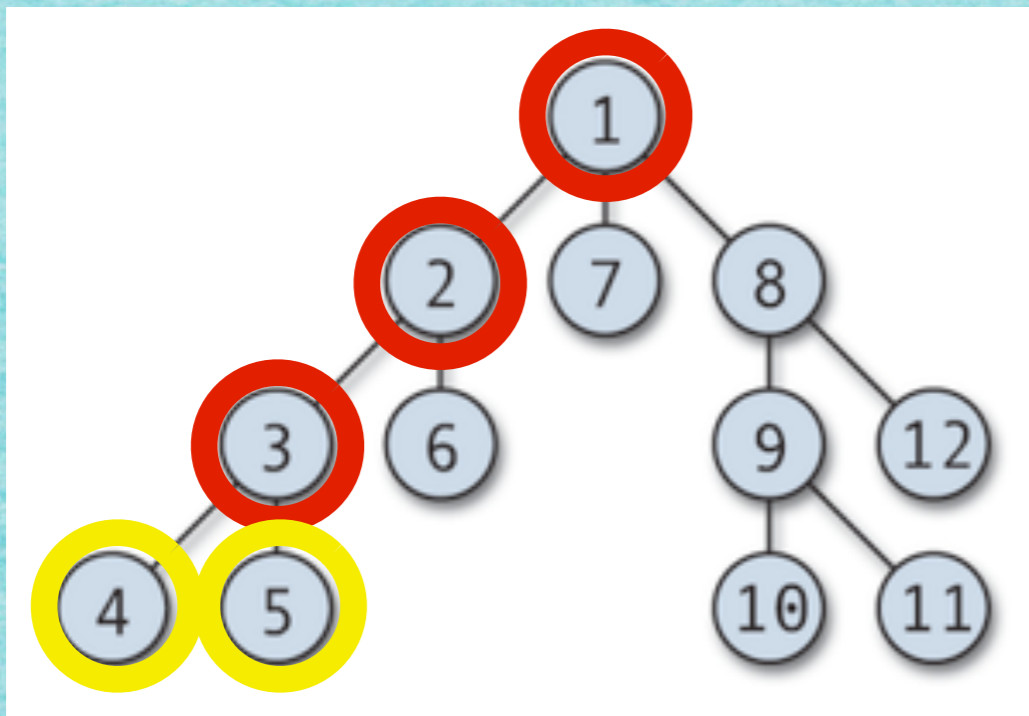
R: 1,2,3

# Graphenscan mit STAPEL



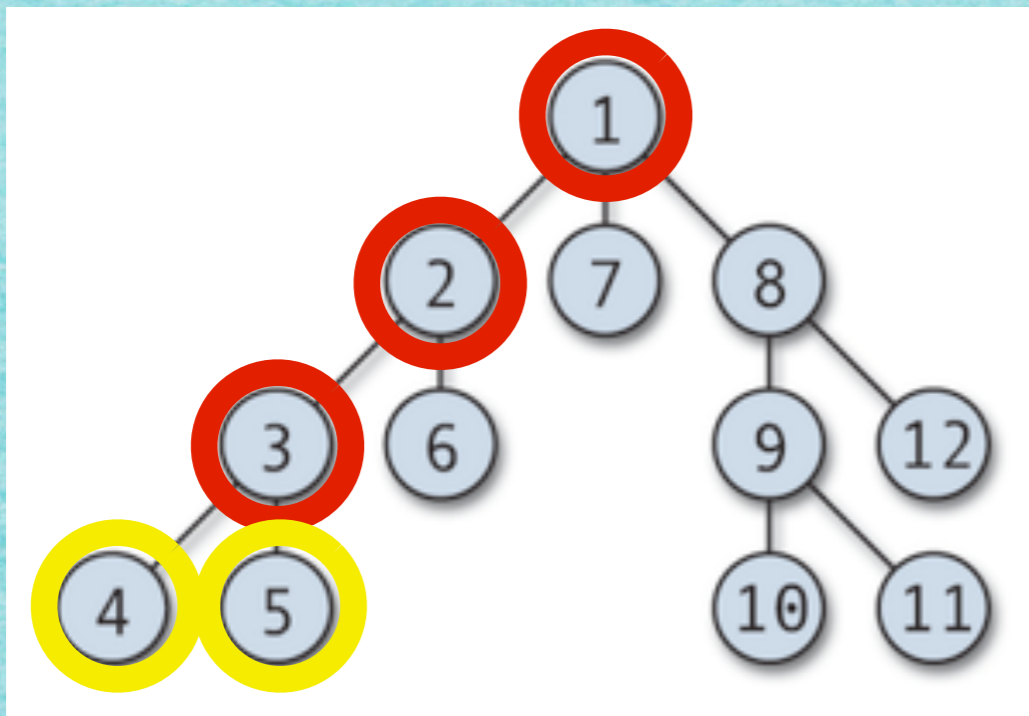
R: 1,2,3,5

# Graphenscan mit STAPEL



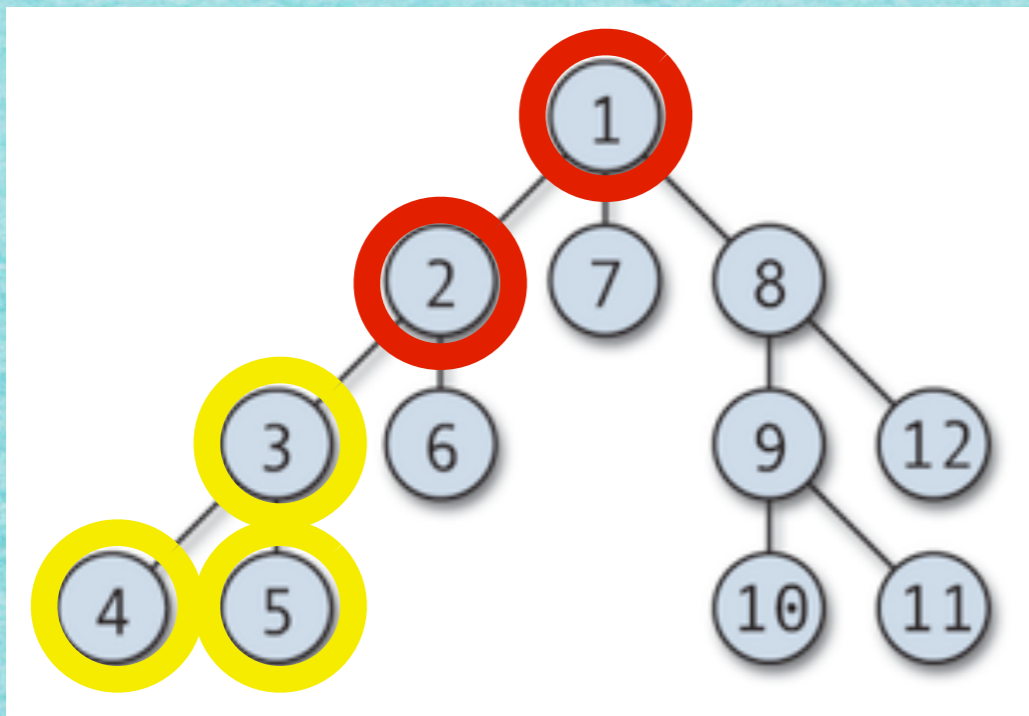
R: 1,2,3,5

# Graphenscan mit STAPEL



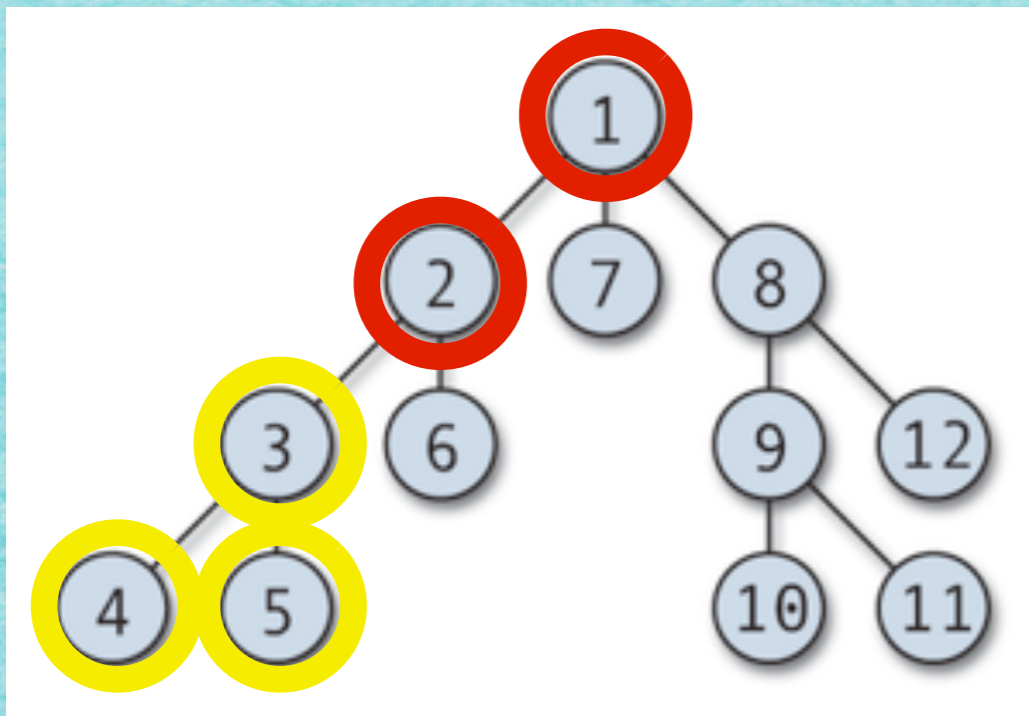
R: 1,2,3,

# Graphenscan mit STAPEL



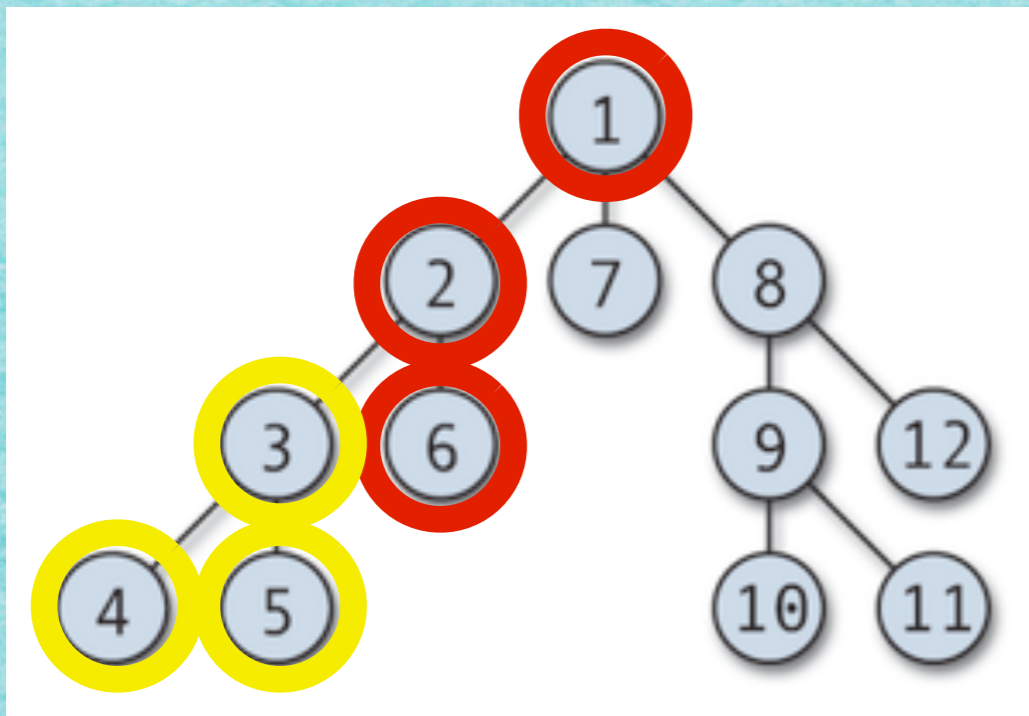
R: 1,2,3,

# Graphenscan mit STAPEL



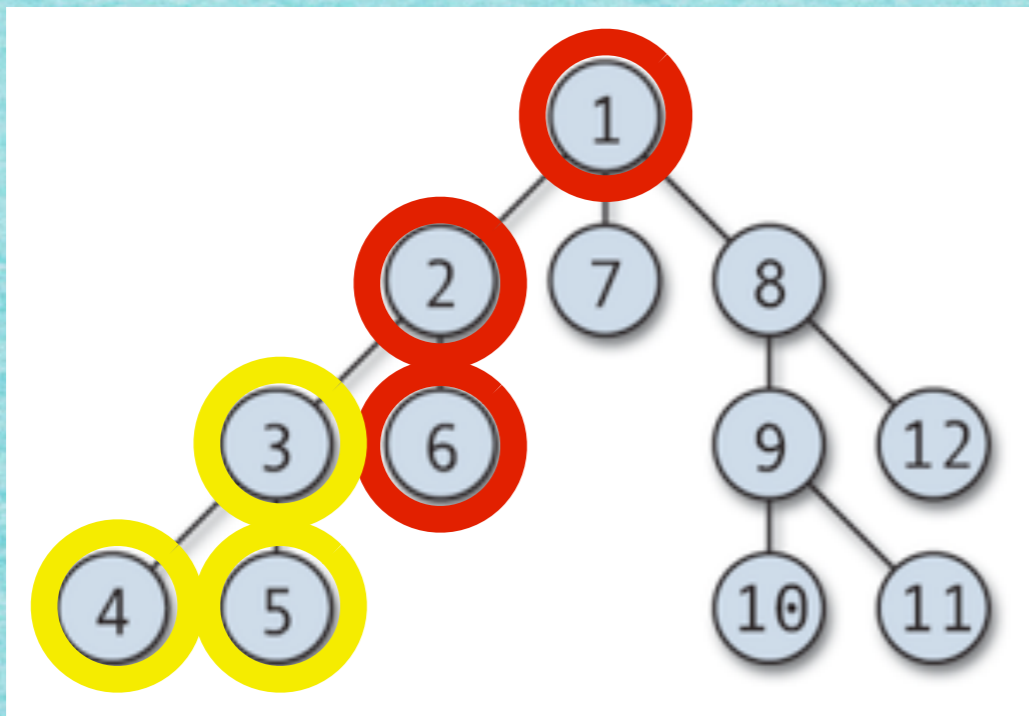
R: 1,2

# Graphenscan mit STAPEL



R: 1,2

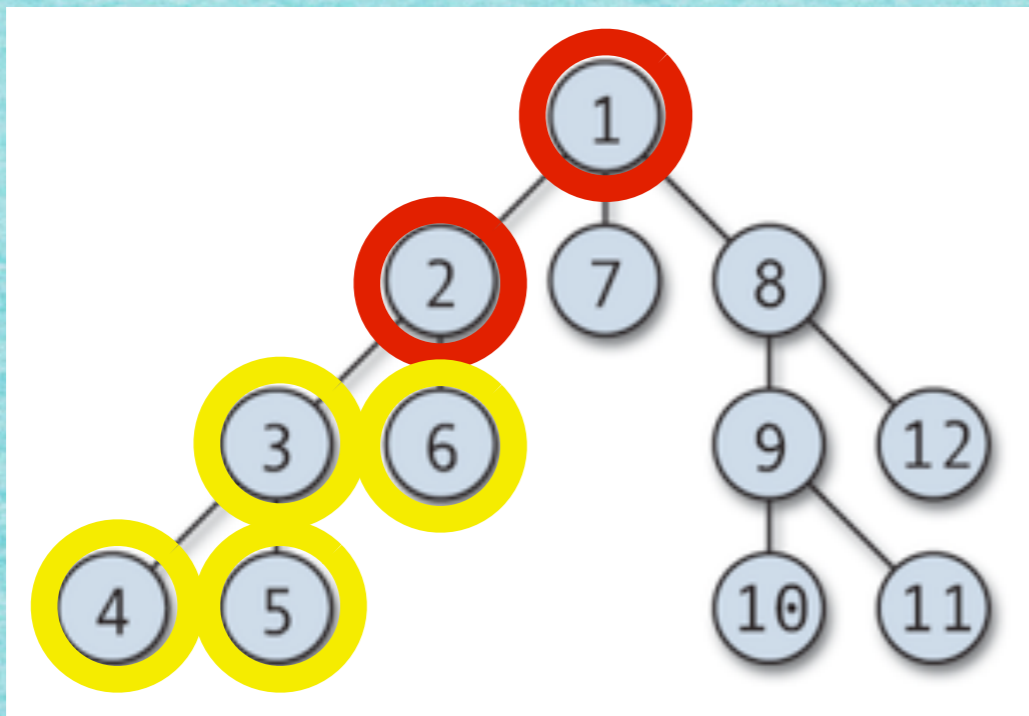
# Graphenscan mit STAPEL



R: 1,2,6

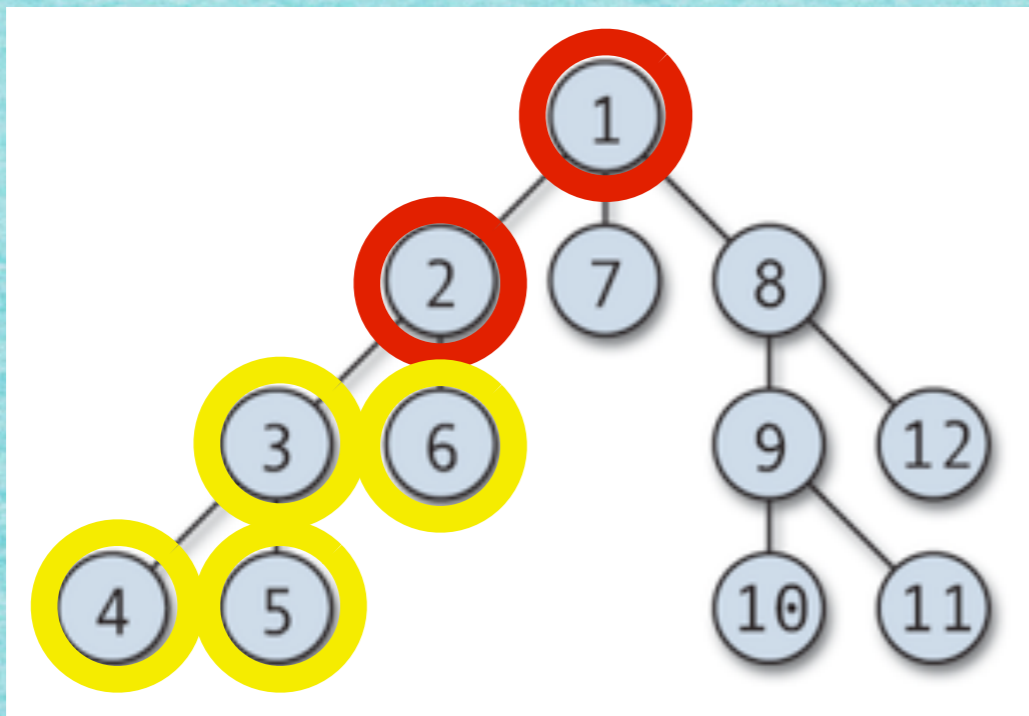


# Graphenscan mit STAPEL



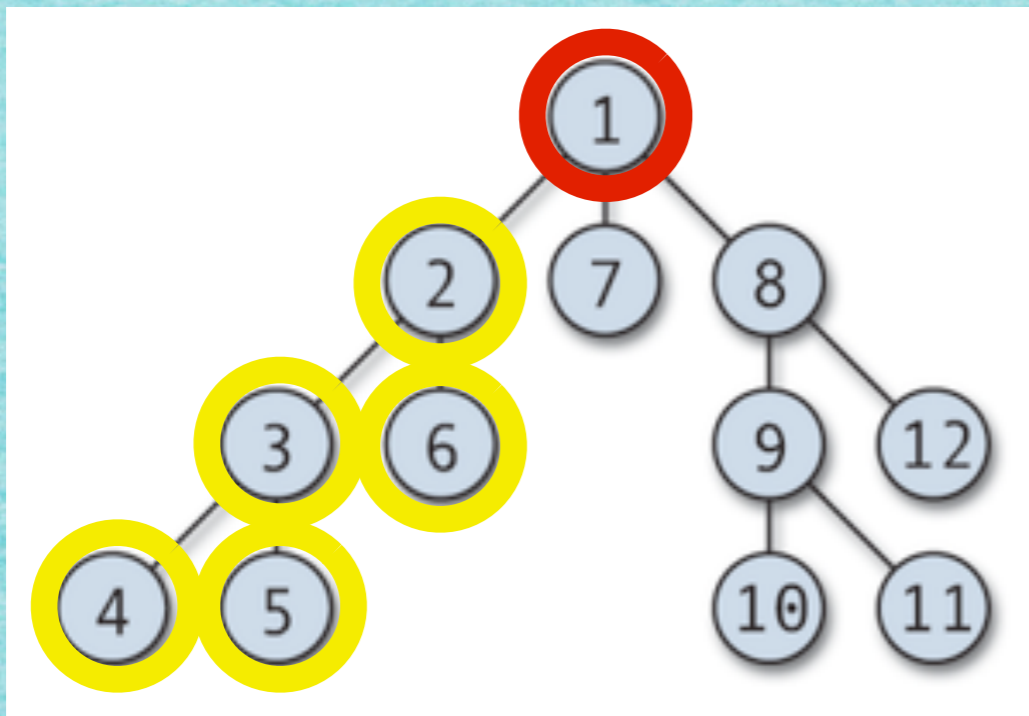
R: 1,2,6

# Graphenscan mit STAPEL



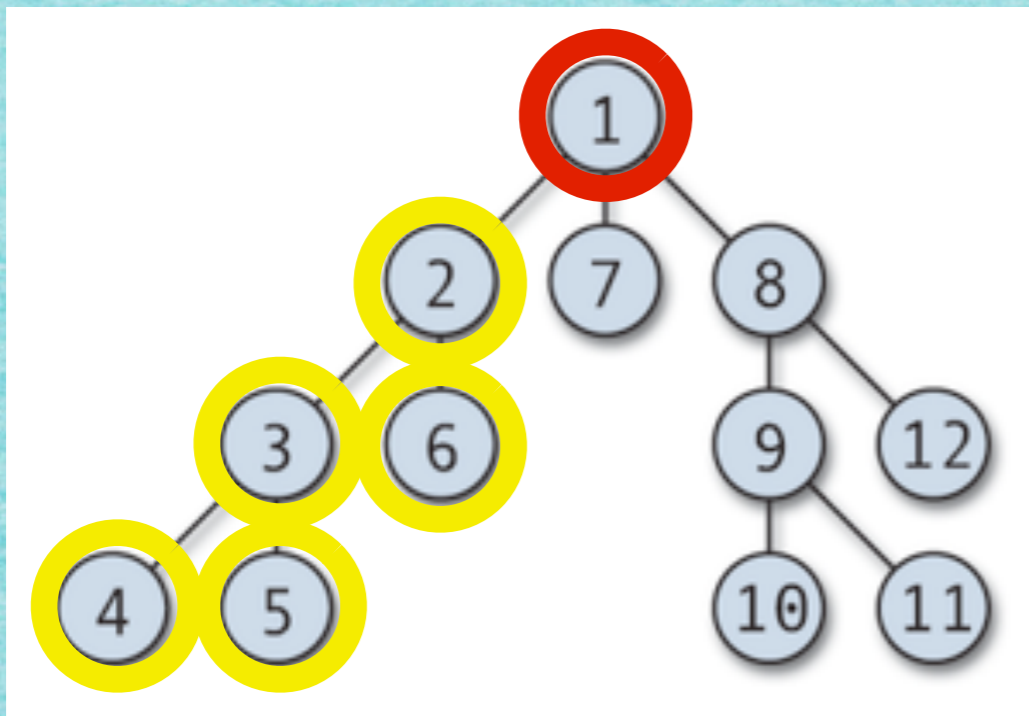
R: 1,2

# Graphenscan mit STAPEL



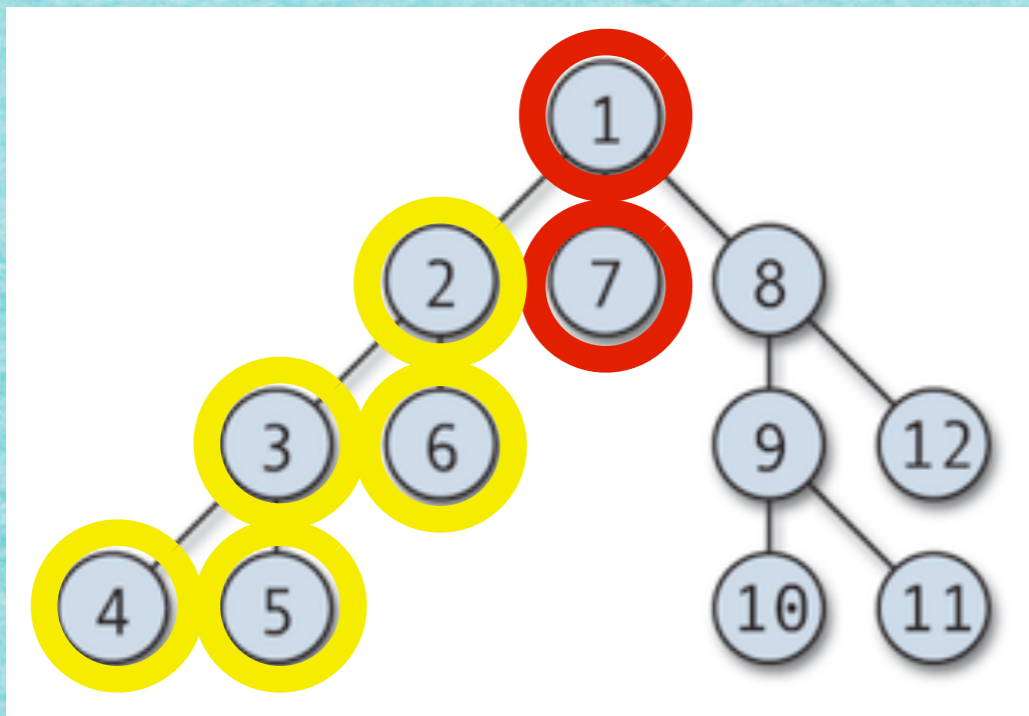
R: 1,2

# Graphenscan mit STAPEL



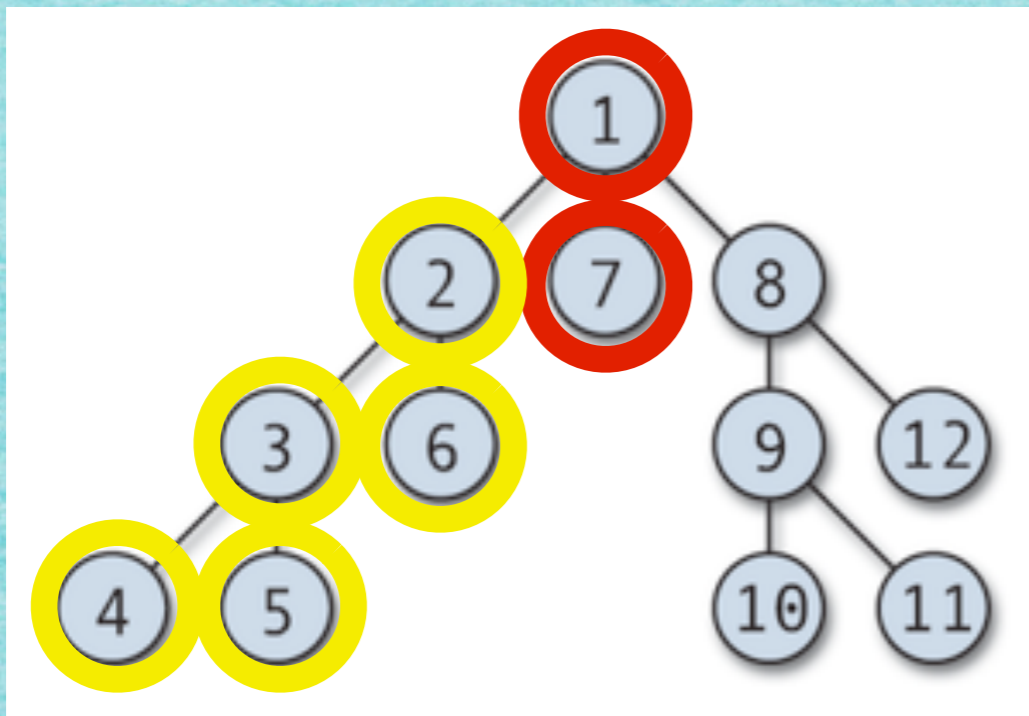
R: 1

# Graphenscan mit STAPEL



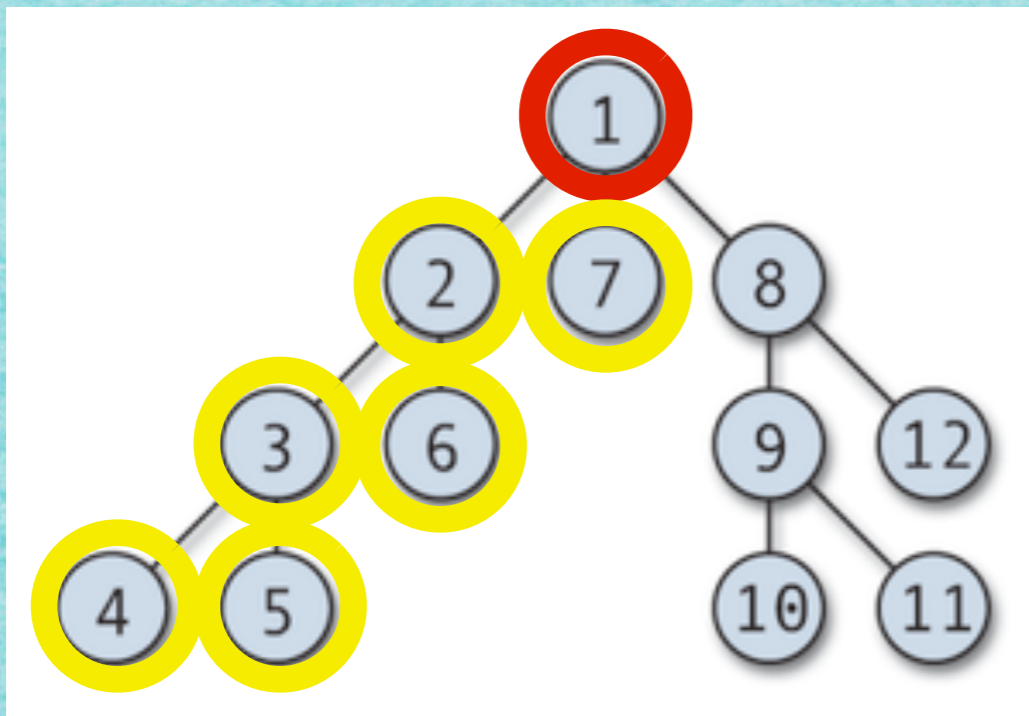
R: 1

# Graphenscan mit STAPEL



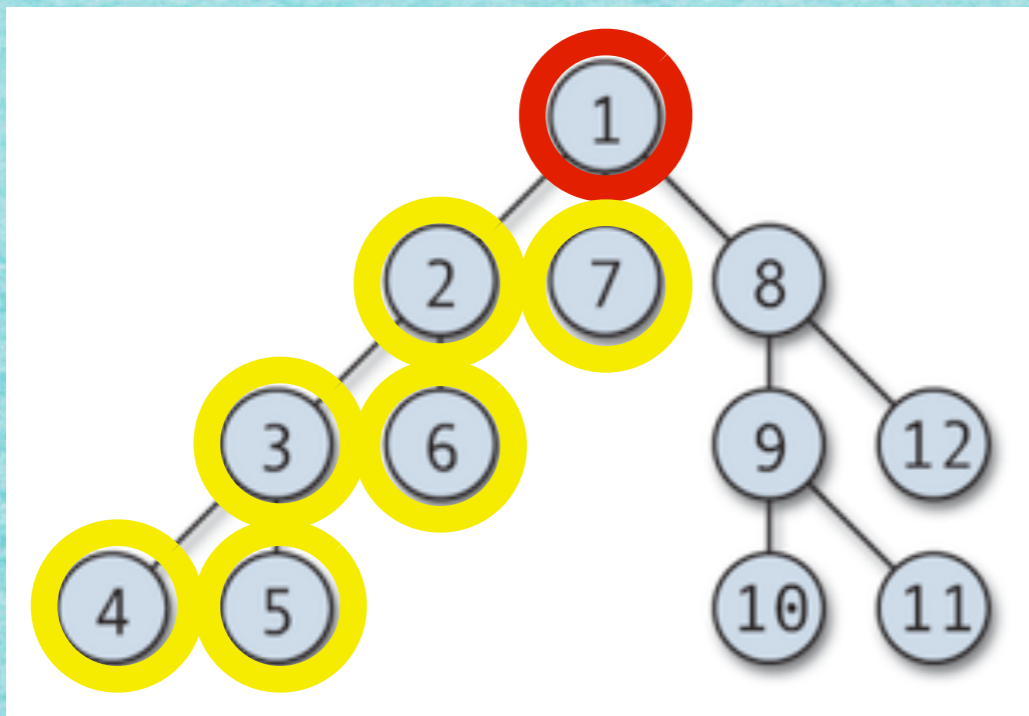
R: 1,7

# Graphenscan mit STAPEL



R: 1,7

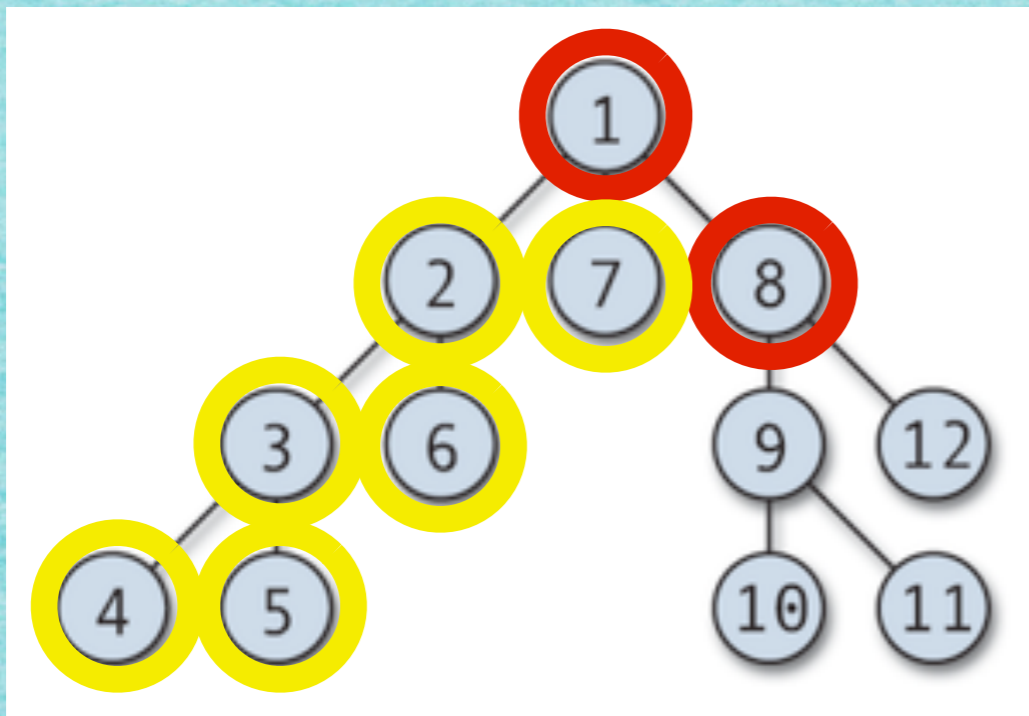
# Graphenscan mit STAPEL



R: 1

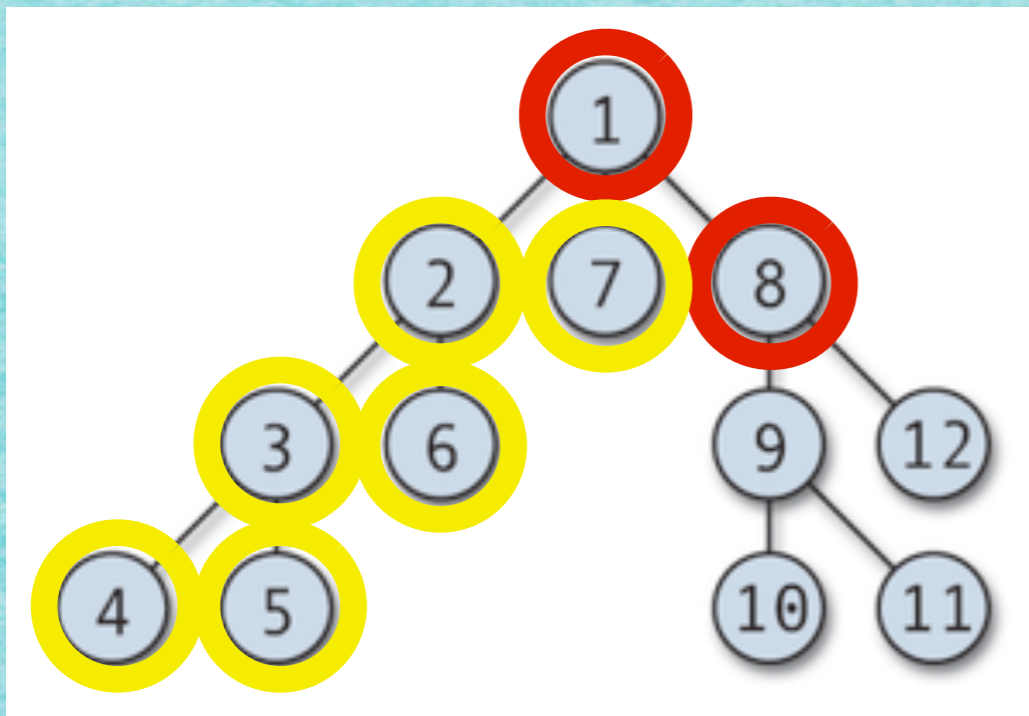


# Graphenscan mit STAPEL



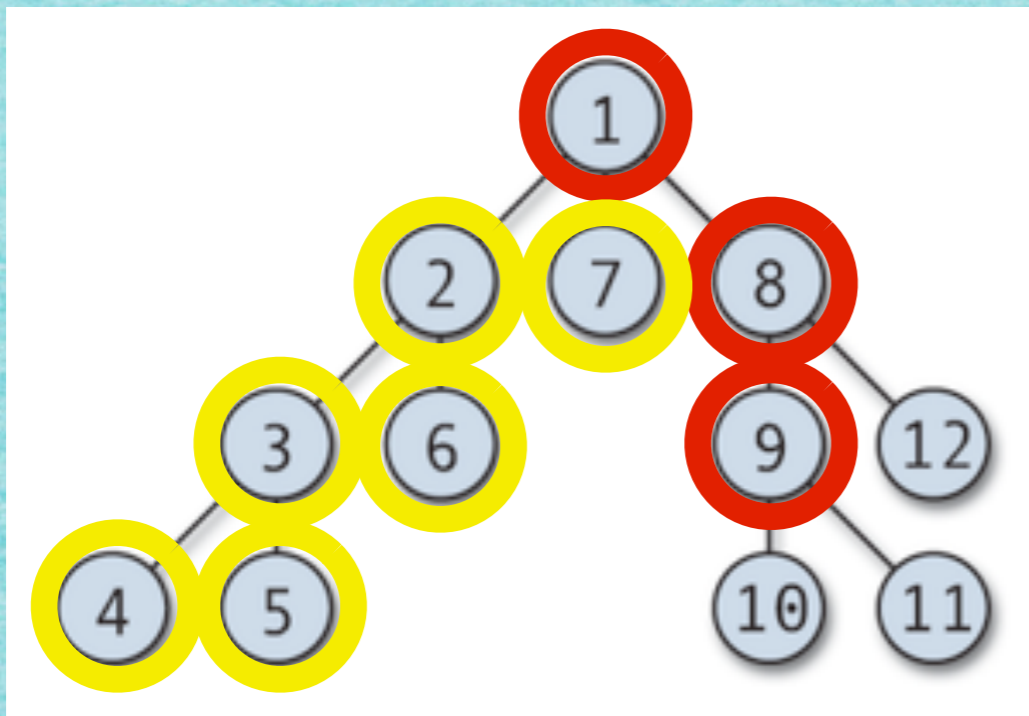
R: 1

# Graphenscan mit STAPEL



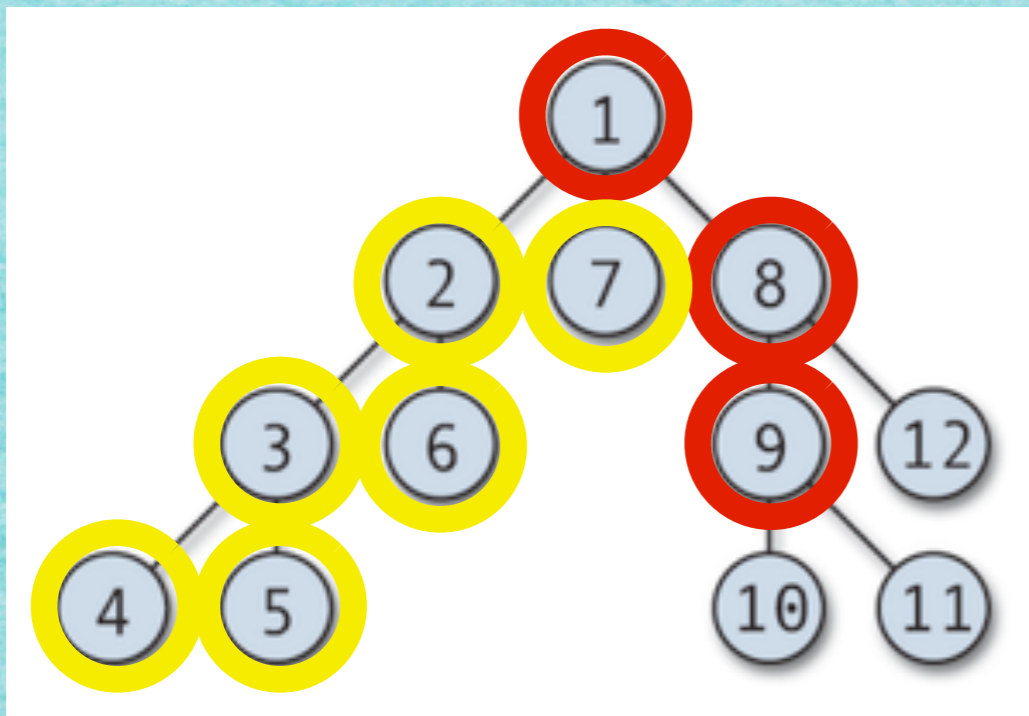
R: 1,8

# Graphenscan mit STAPEL



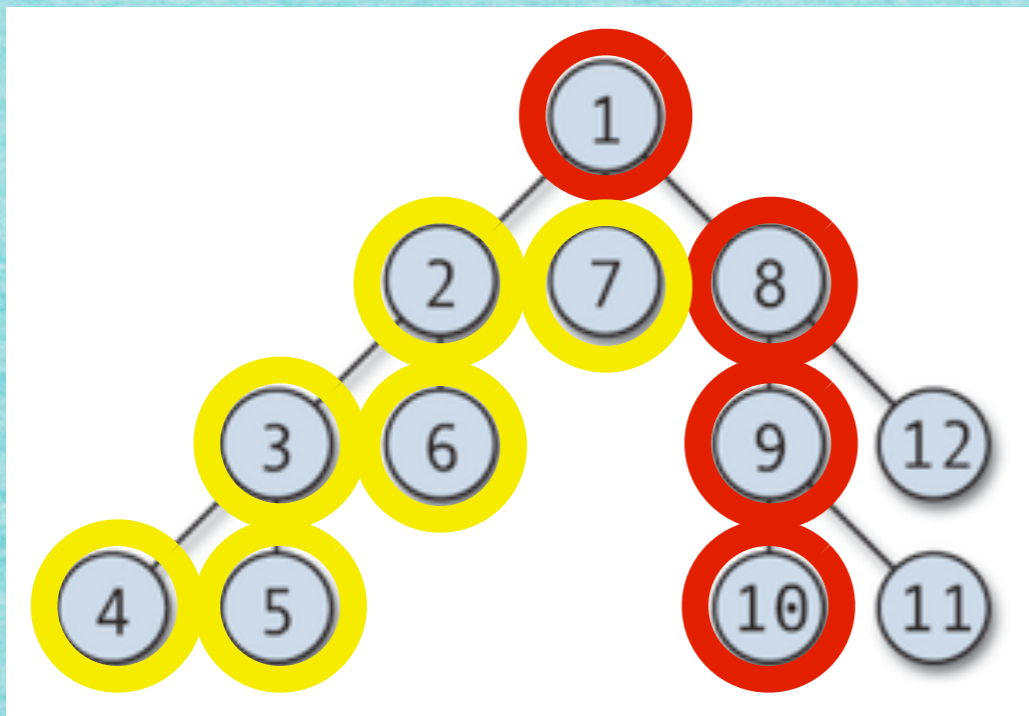
R: 1,8

# Graphenscan mit STAPEL



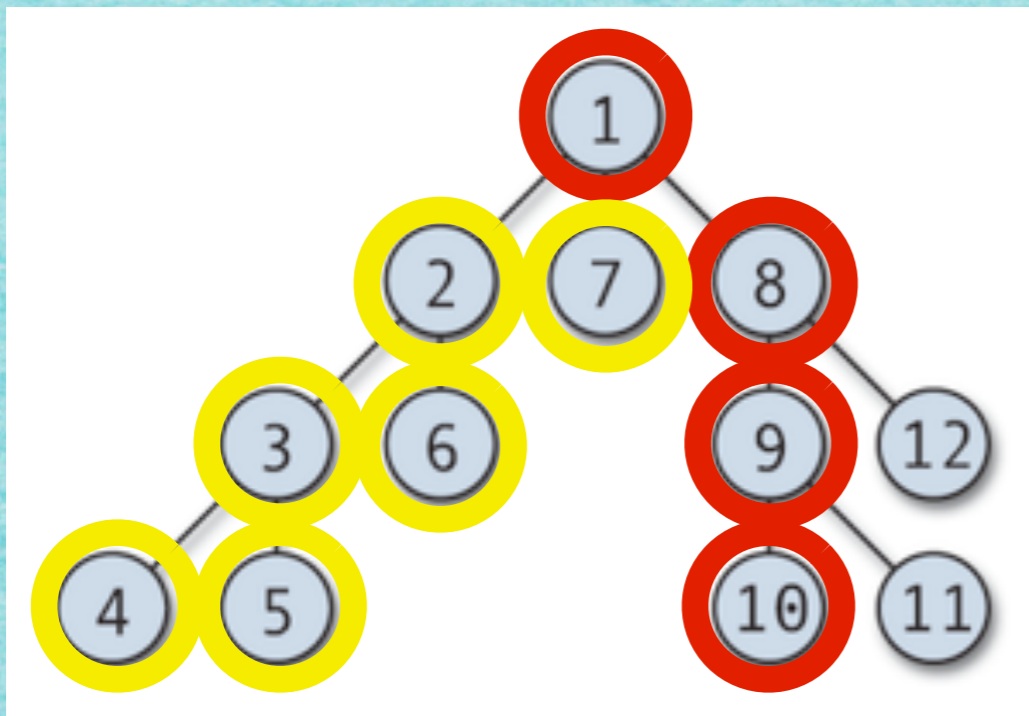
R: 1,8,9

# Graphenscan mit STAPEL



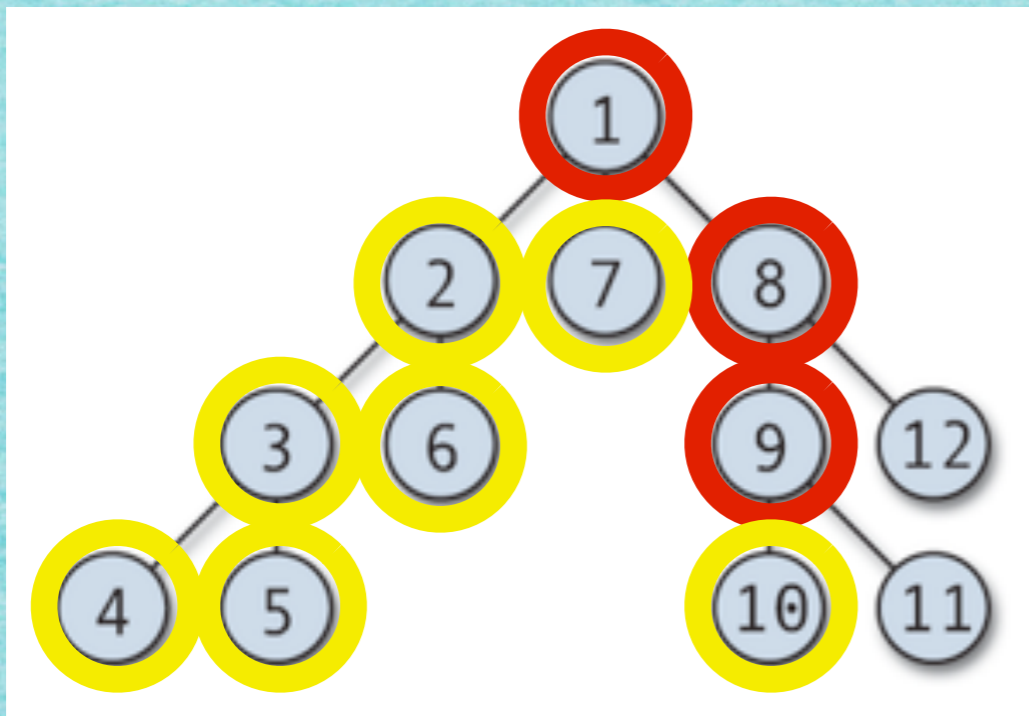
R: 1,8,9

# Graphenscan mit STAPEL



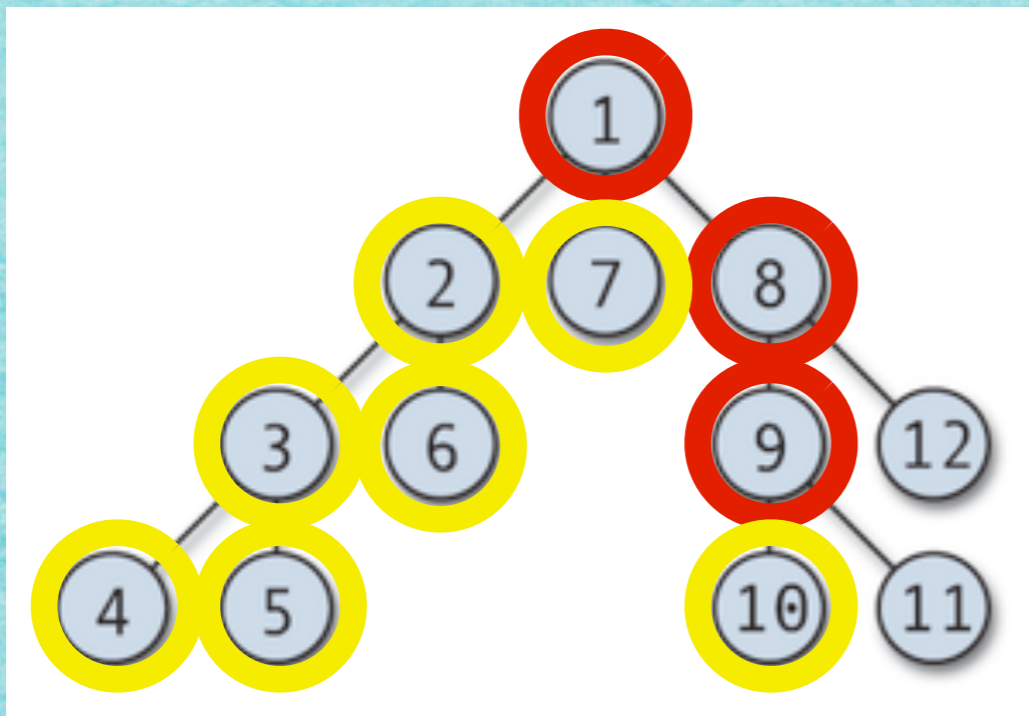
R: 1,8,9,10

# Graphenscan mit STAPEL



R: 1,8,9,10

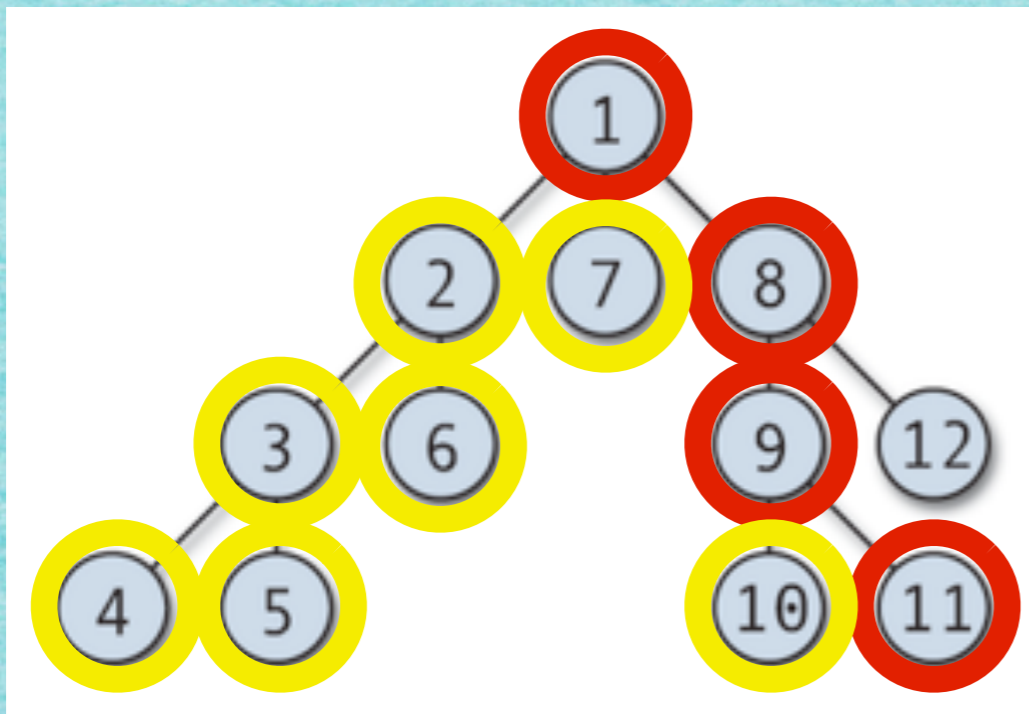
# Graphenscan mit STAPEL



R: 1,8,9

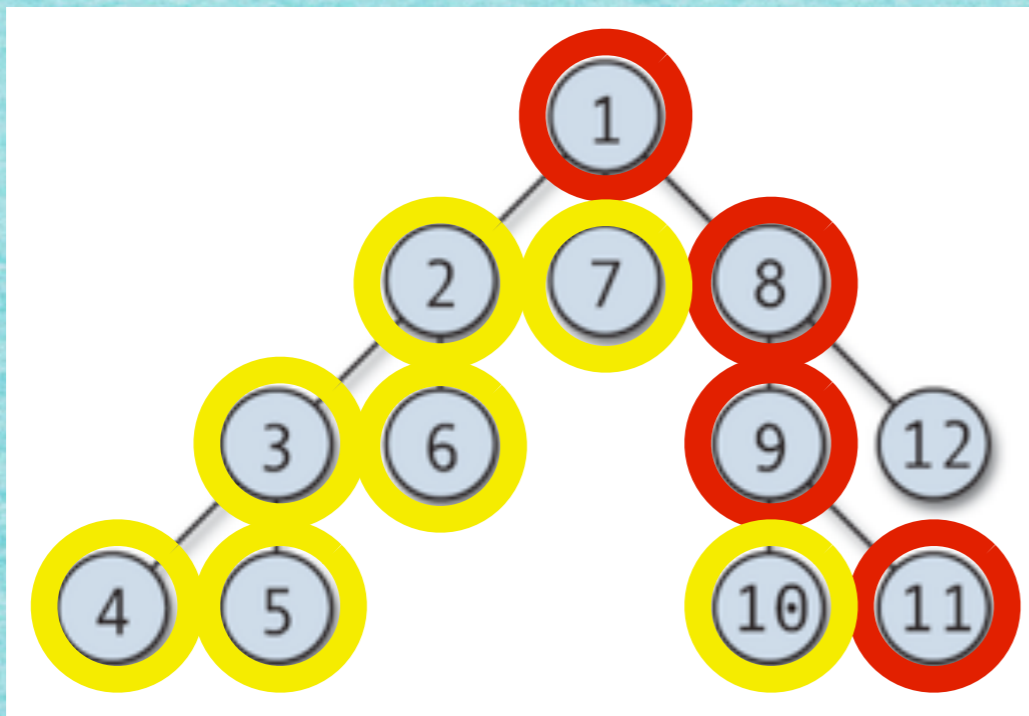


# Graphenscan mit STAPEL



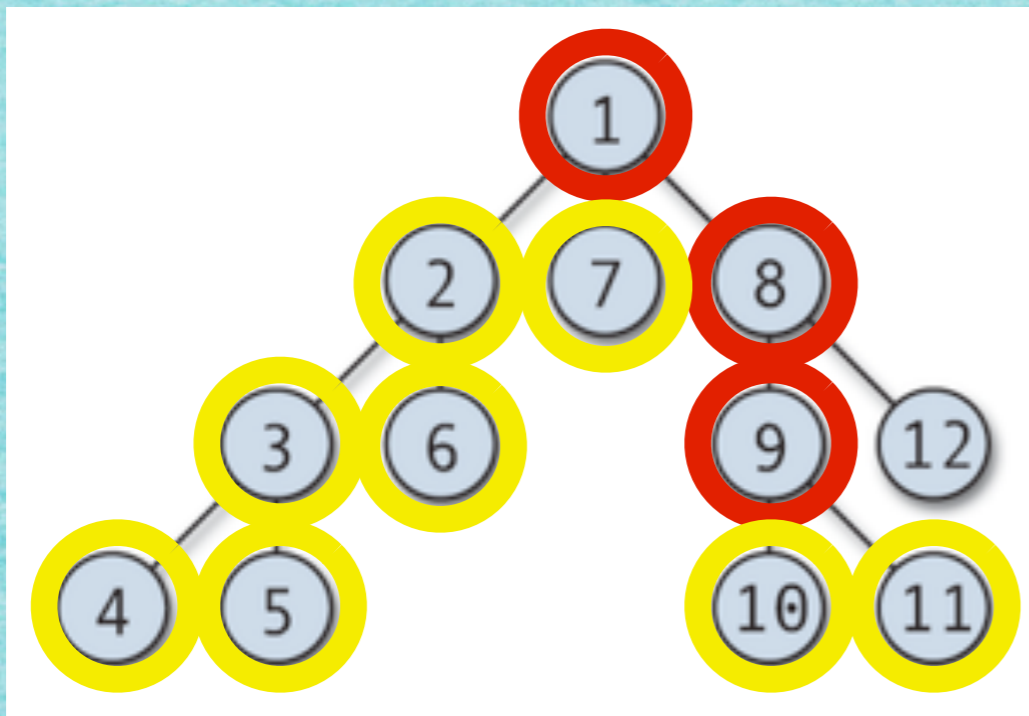
R: 1,8,9

# Graphenscan mit STAPEL



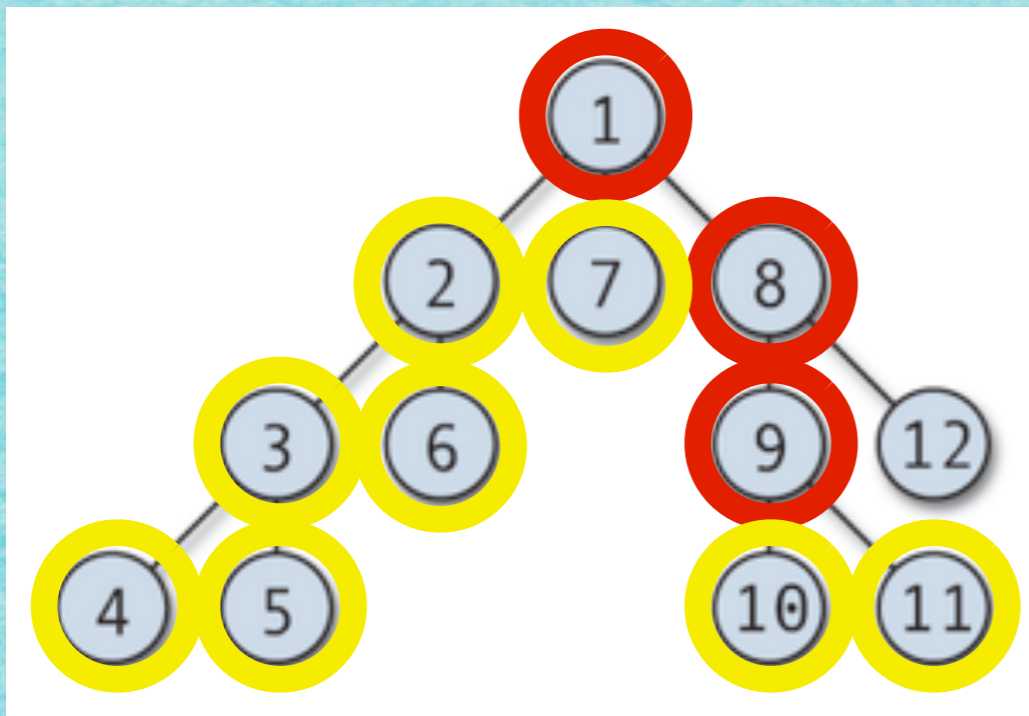
R: 1,8,9,11

# Graphenscan mit STAPEL



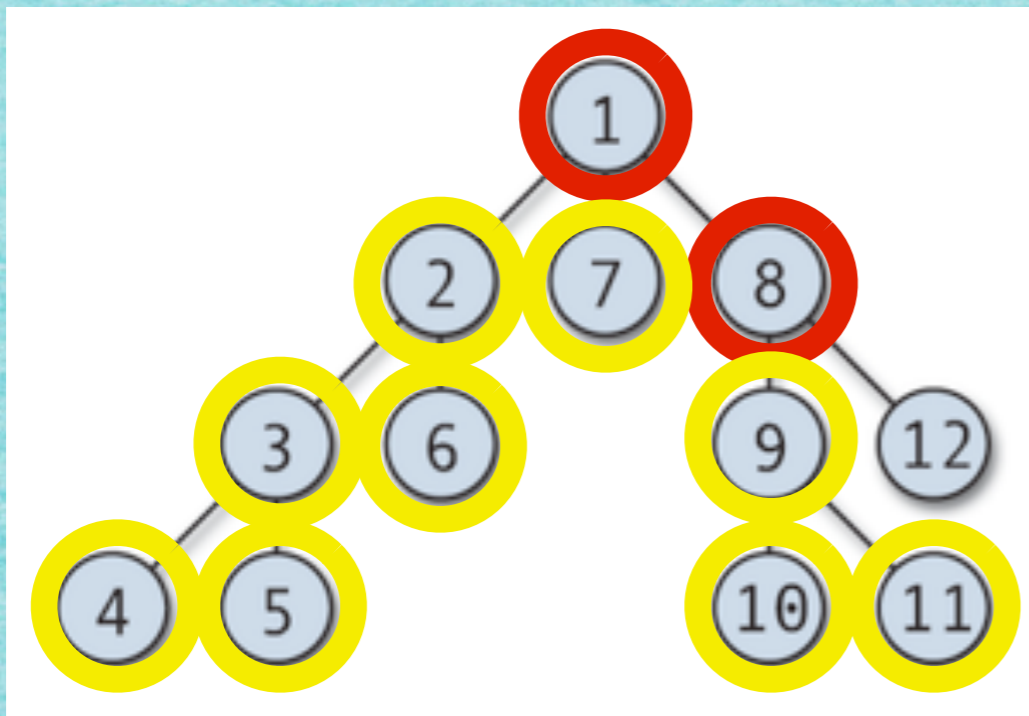
R: 1,8,9,11

# Graphenscan mit STAPEL



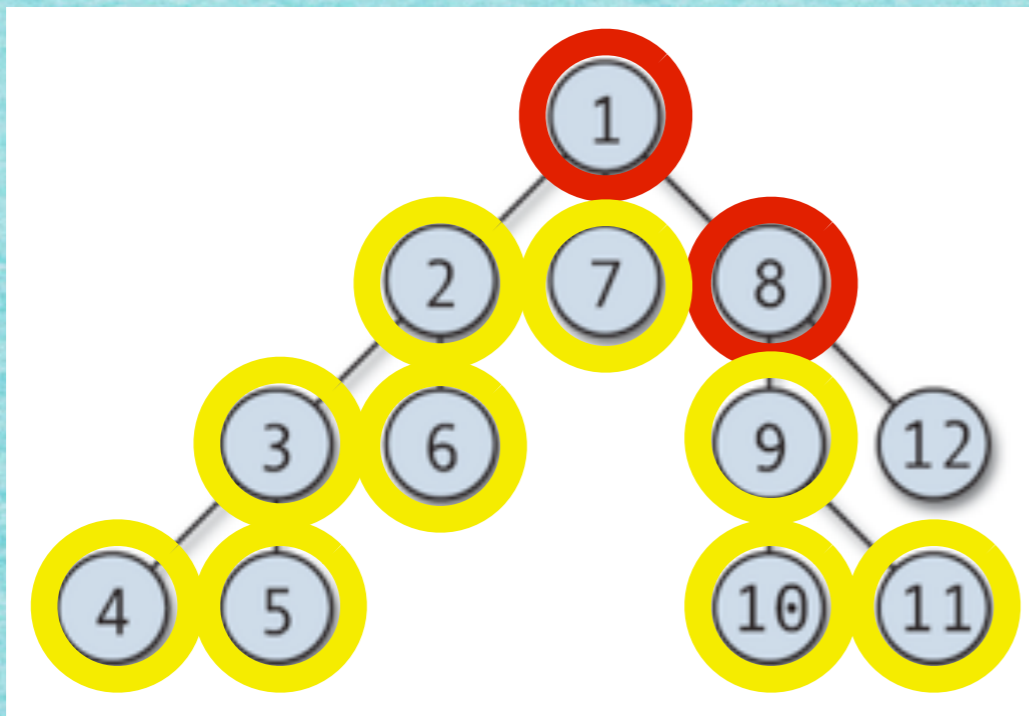
R: 1,8,9

# Graphenscan mit STAPEL



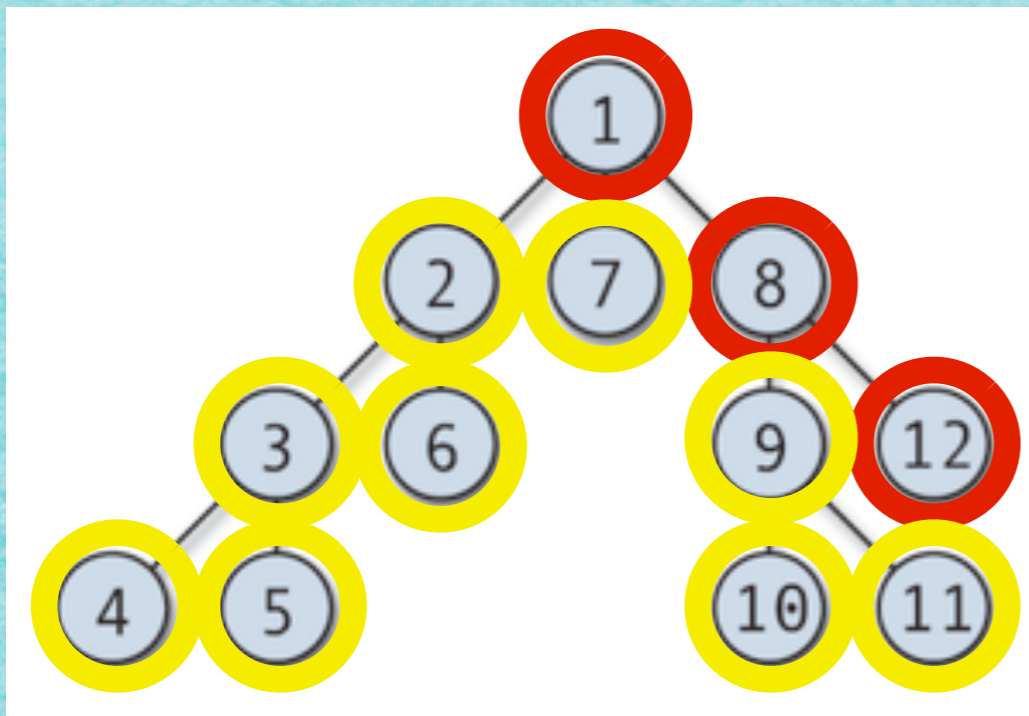
R: 1,8,9

# Graphenscan mit STAPEL



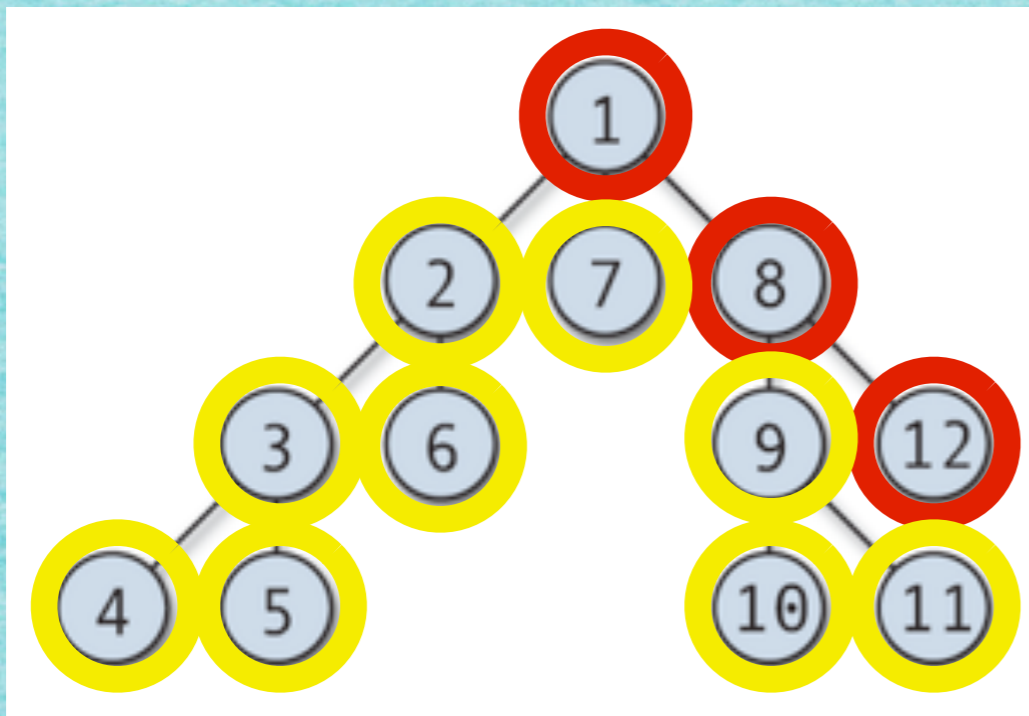
R: 1,8

# Graphenscan mit STAPEL



R: 1,8

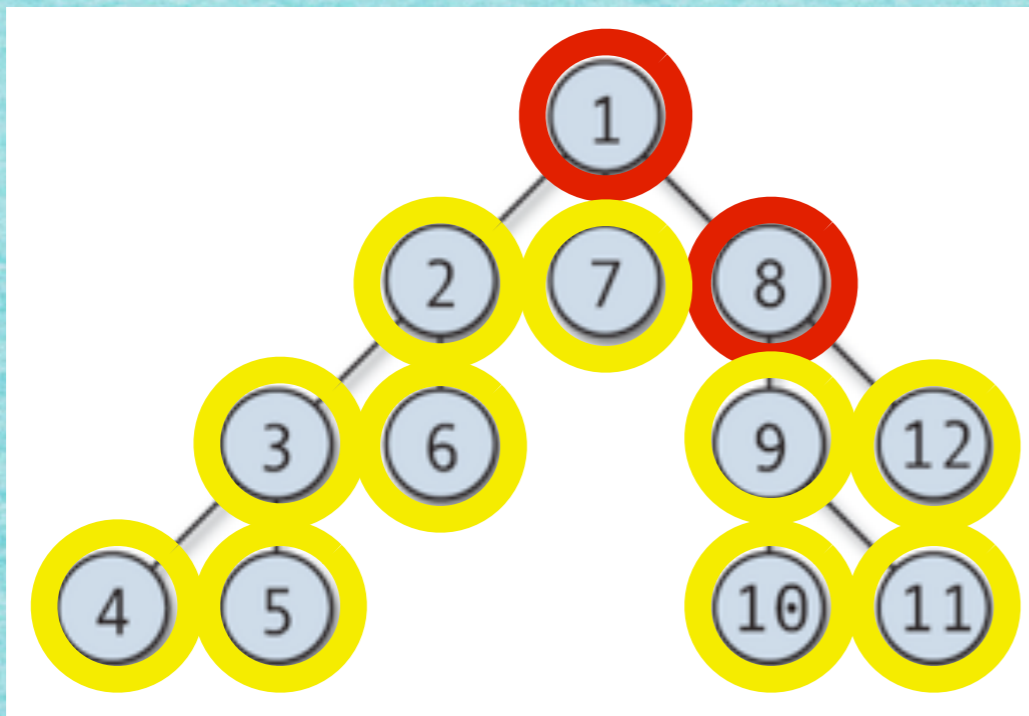
# Graphenscan mit STAPEL



R: 1,8,12

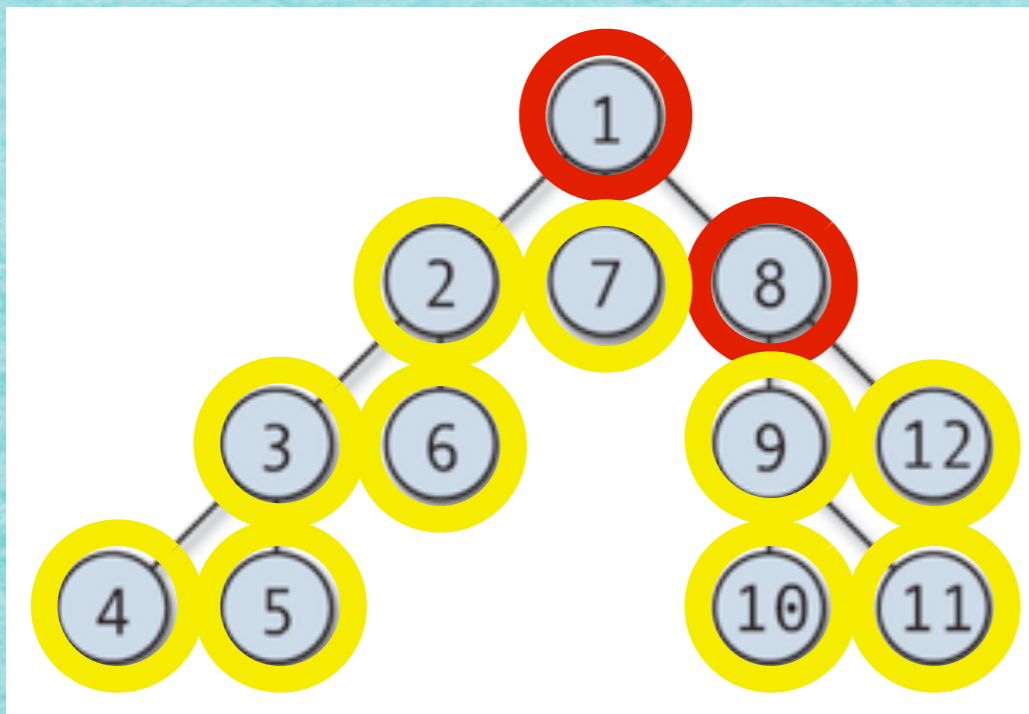


# Graphenscan mit STAPEL



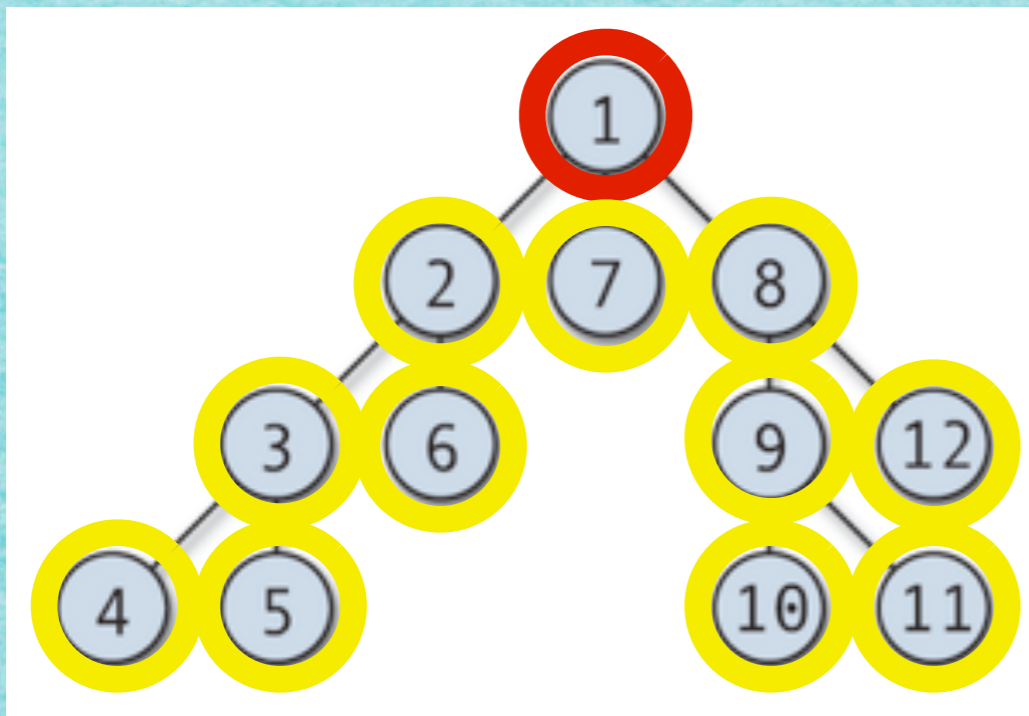
R: 1,8,12

# Graphenscan mit STAPEL



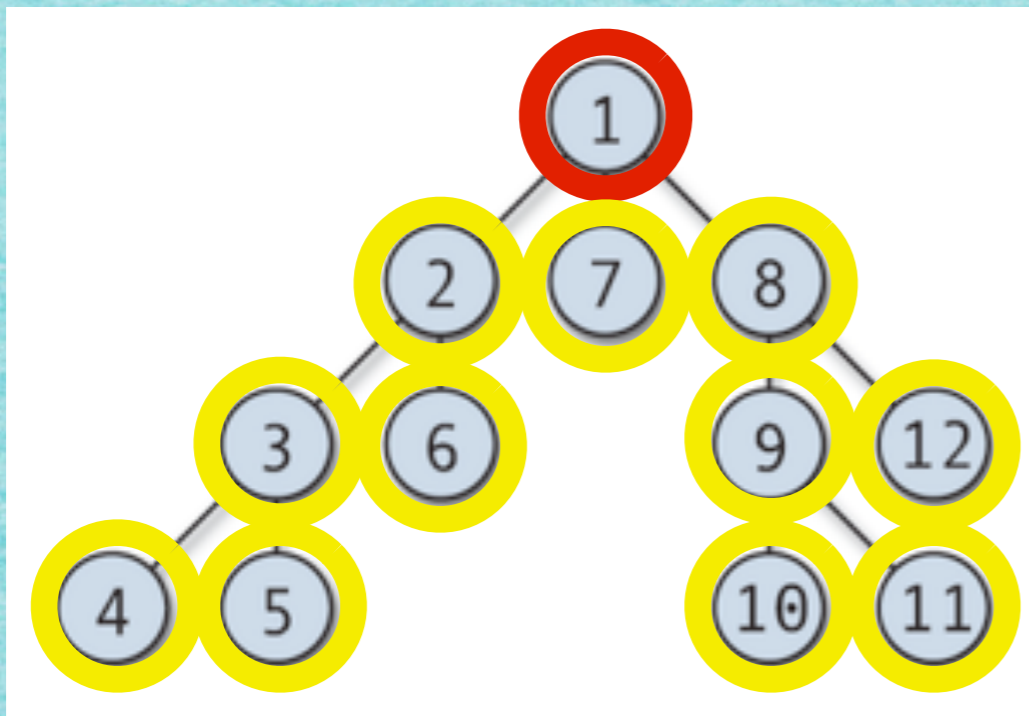
R: 1,8

# Graphenscan mit STAPEL



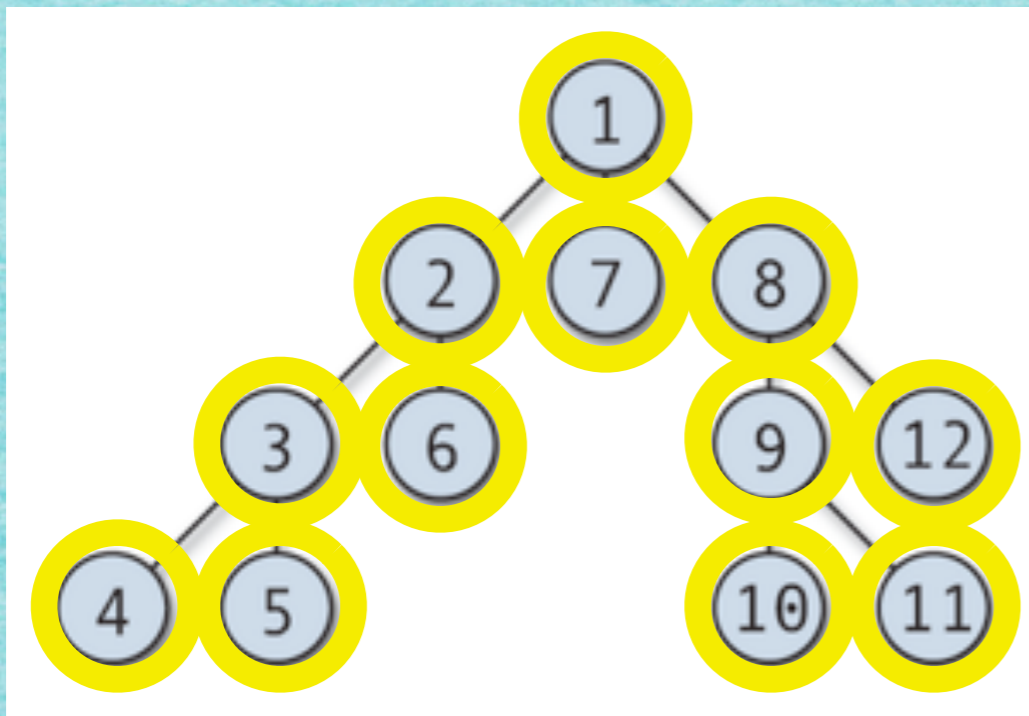
R: 1,8

# Graphenscan mit STAPEL



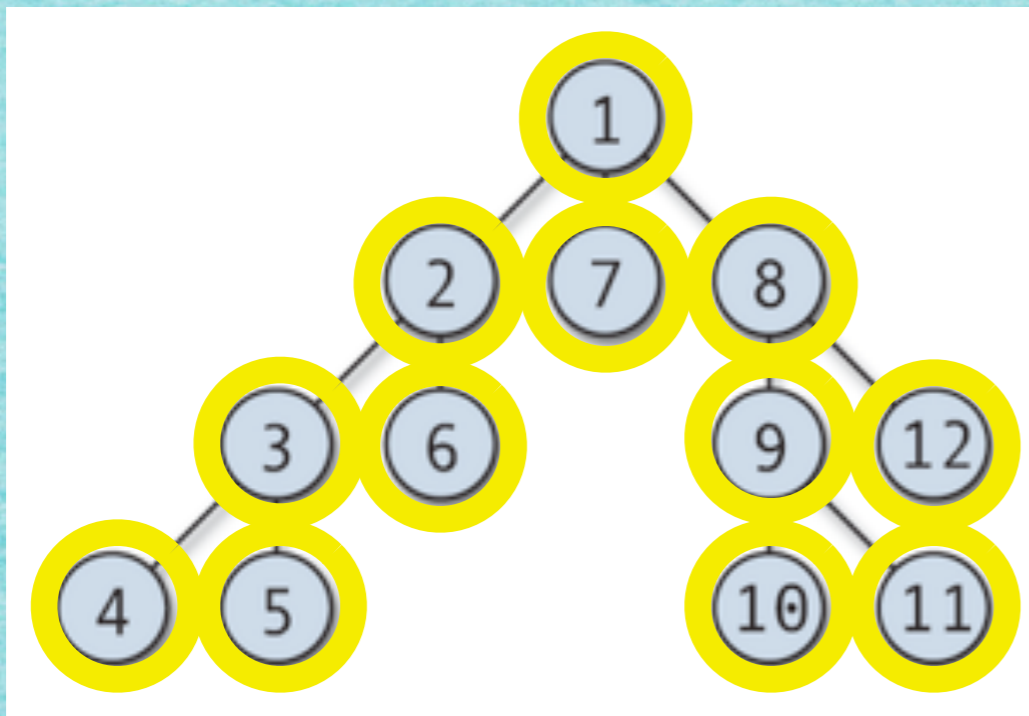
R: 1

# Graphenscan mit STAPEL



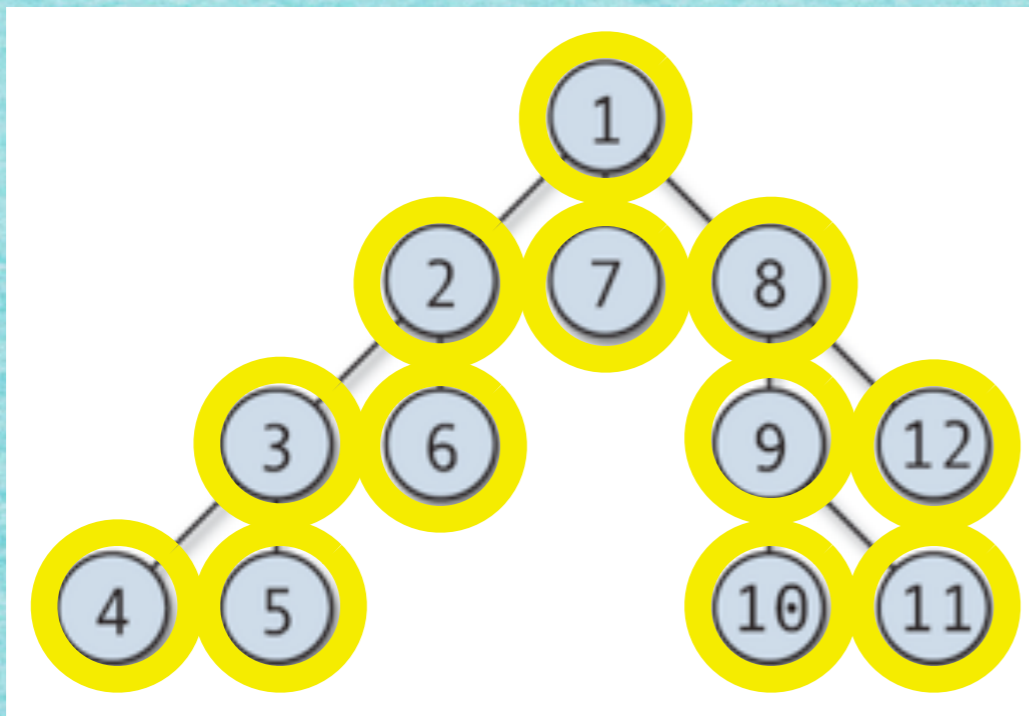
R: 1

# Graphenscan mit STAPEL



R:

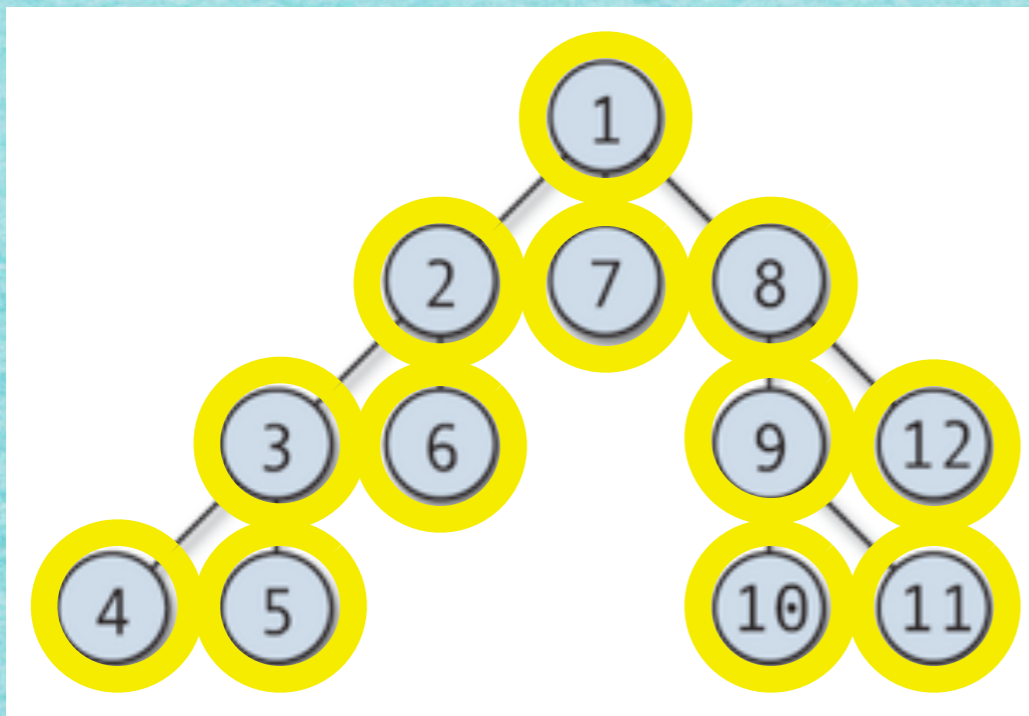
# Graphenscan mit STAPEL



R:

**STOP!**

# Graphenscan mit STAPEL



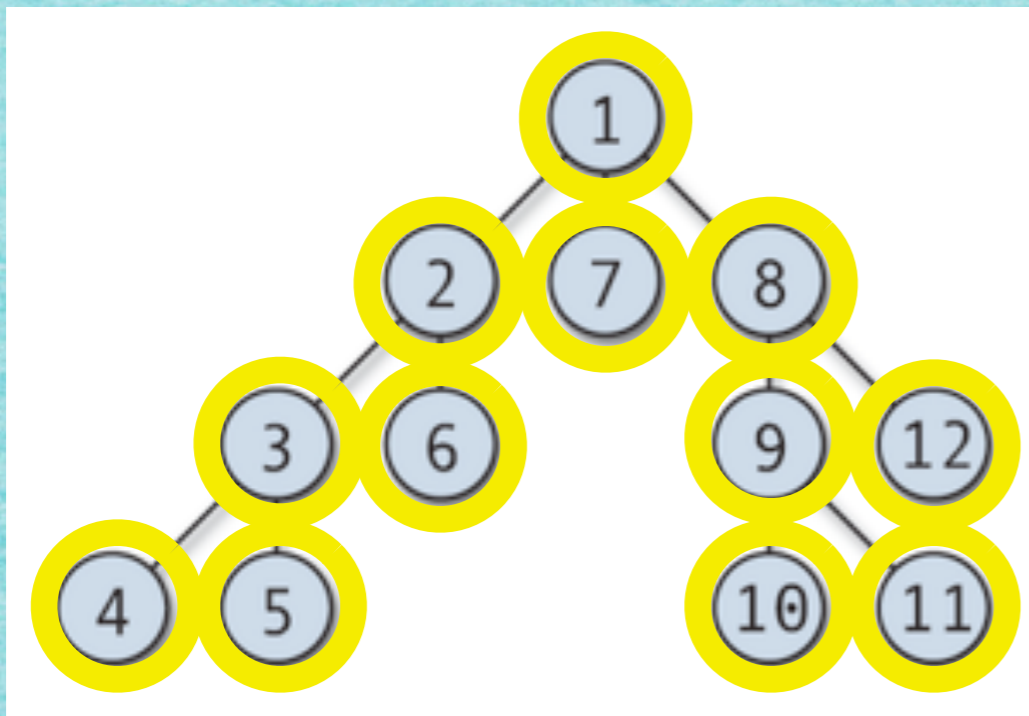
R:

**STOP!**

**TIEFENSUCHE - "DEPTH-FIRST SEARCH" (DFS)**



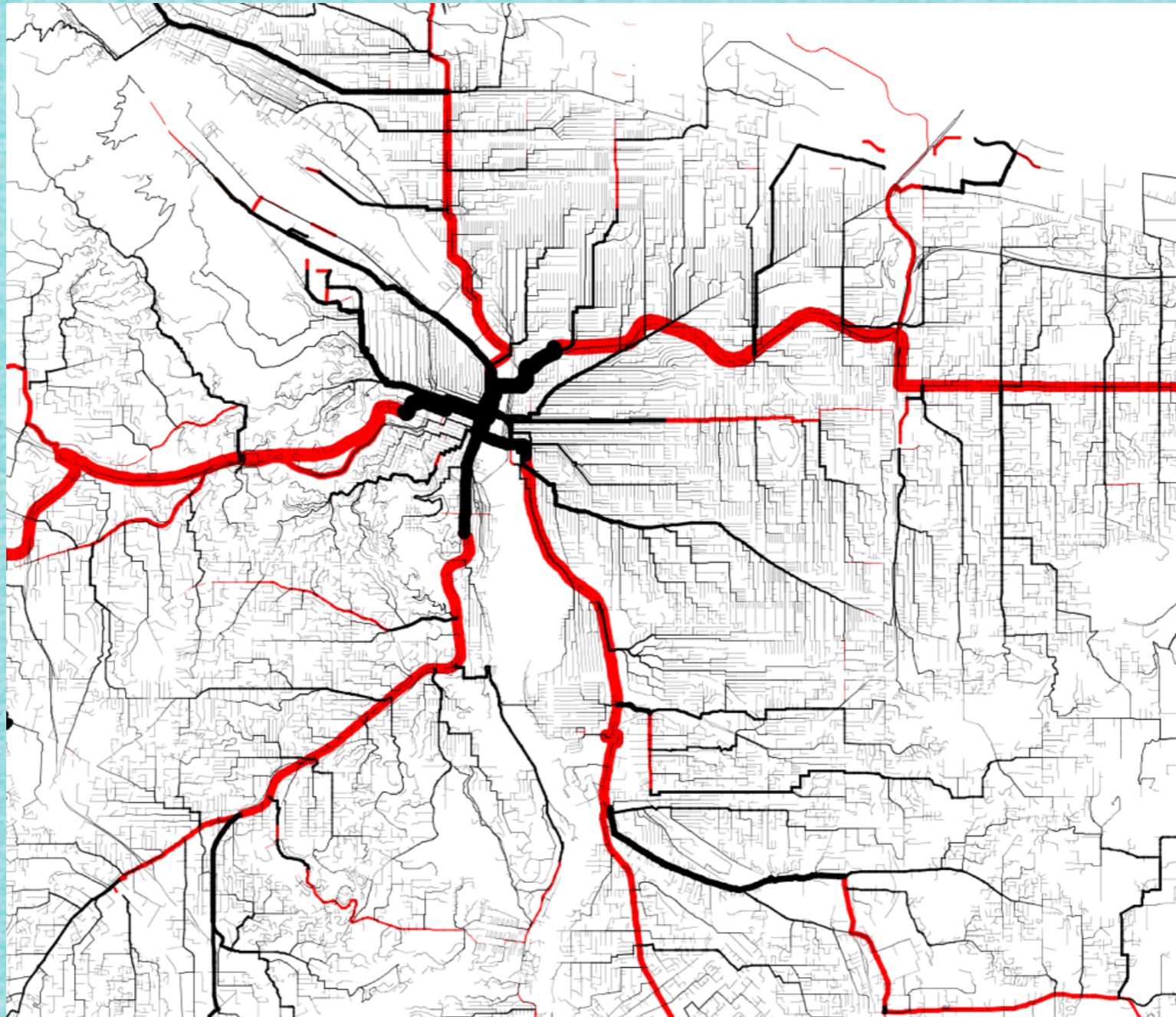
# Graphenscan mit STAPEL



R:

**STOP!**

**Breitensuche liefert kürzeste Wege von einer Quelle aus**



**Viele "Sammler"**

# Auf die Schnelle mit der Welle



# Auf die Schnelle mit der Welle



# Auf die Schnelle mit der Welle

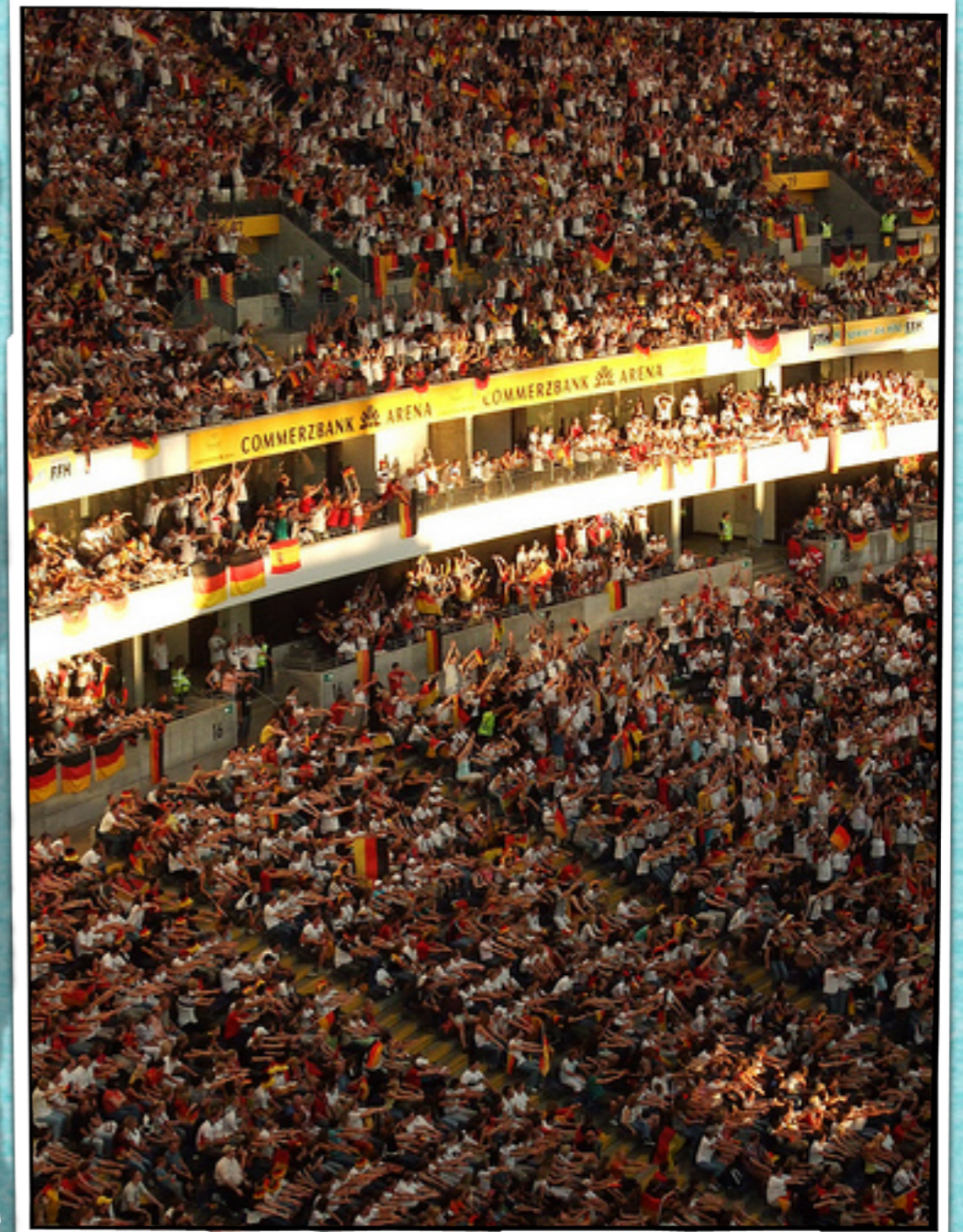
A. LOS bei „NULL“

B. Bis „ANGEKOMMEN!“:

- Solange du noch nicht aufgestanden warst:
  - ▶ Wenn ein oder mehrere direkte Nachbarn aufstehen:
    1. Einen dieser Nachbarn merken
    2. In der nächsten Runde:
      - 2.1. aufstehen
      - 2.2. Zahl merken
    3. In der übernächsten Runde hinsetzen

C. Nach „ANGEKOMMEN!“:

- Auf gemerkten Nachbarn zeigen



# Auf die Schnelle mit der Welle

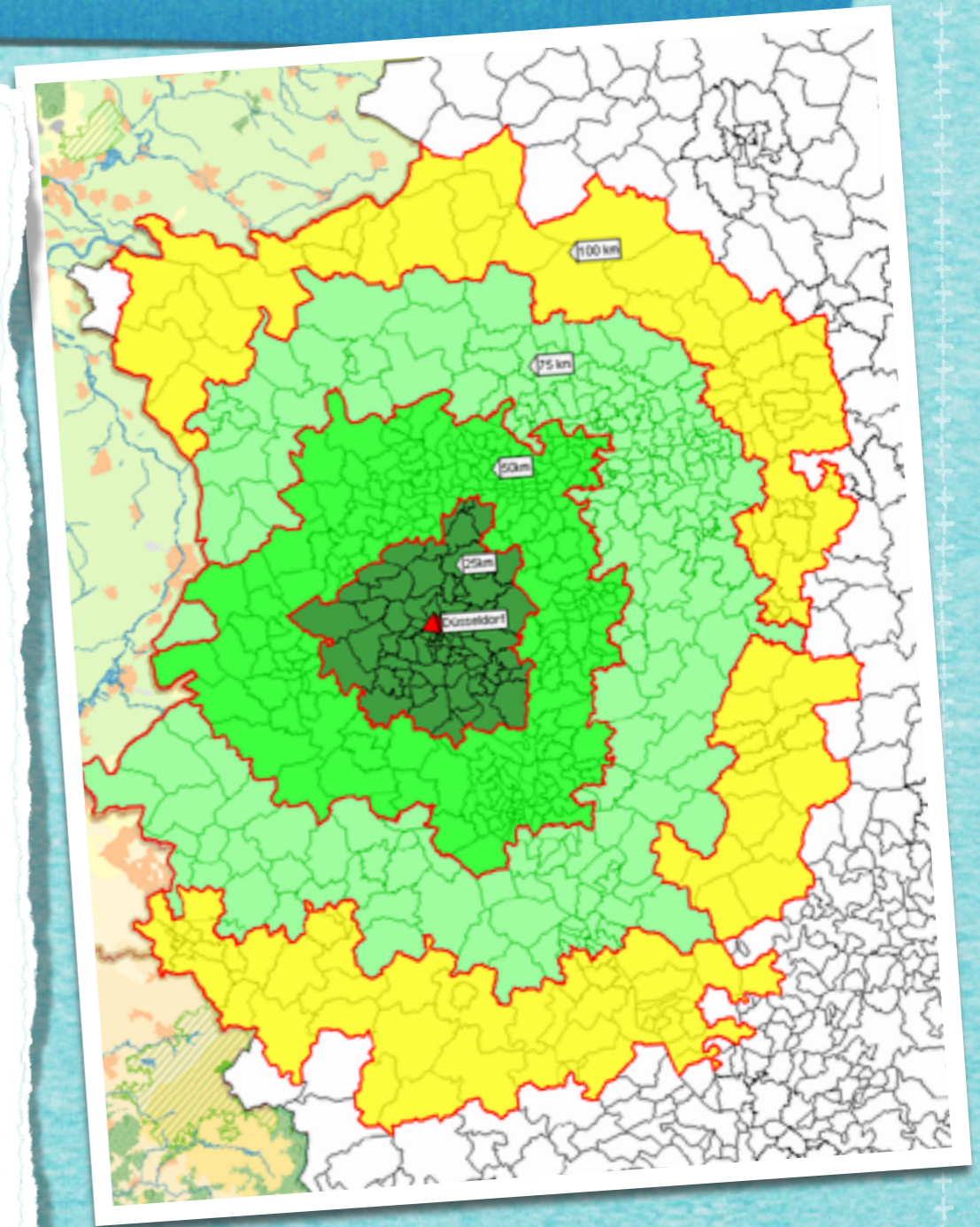
A. LOS bei „NULL“

B. Bis „ANGEKOMMEN!“:

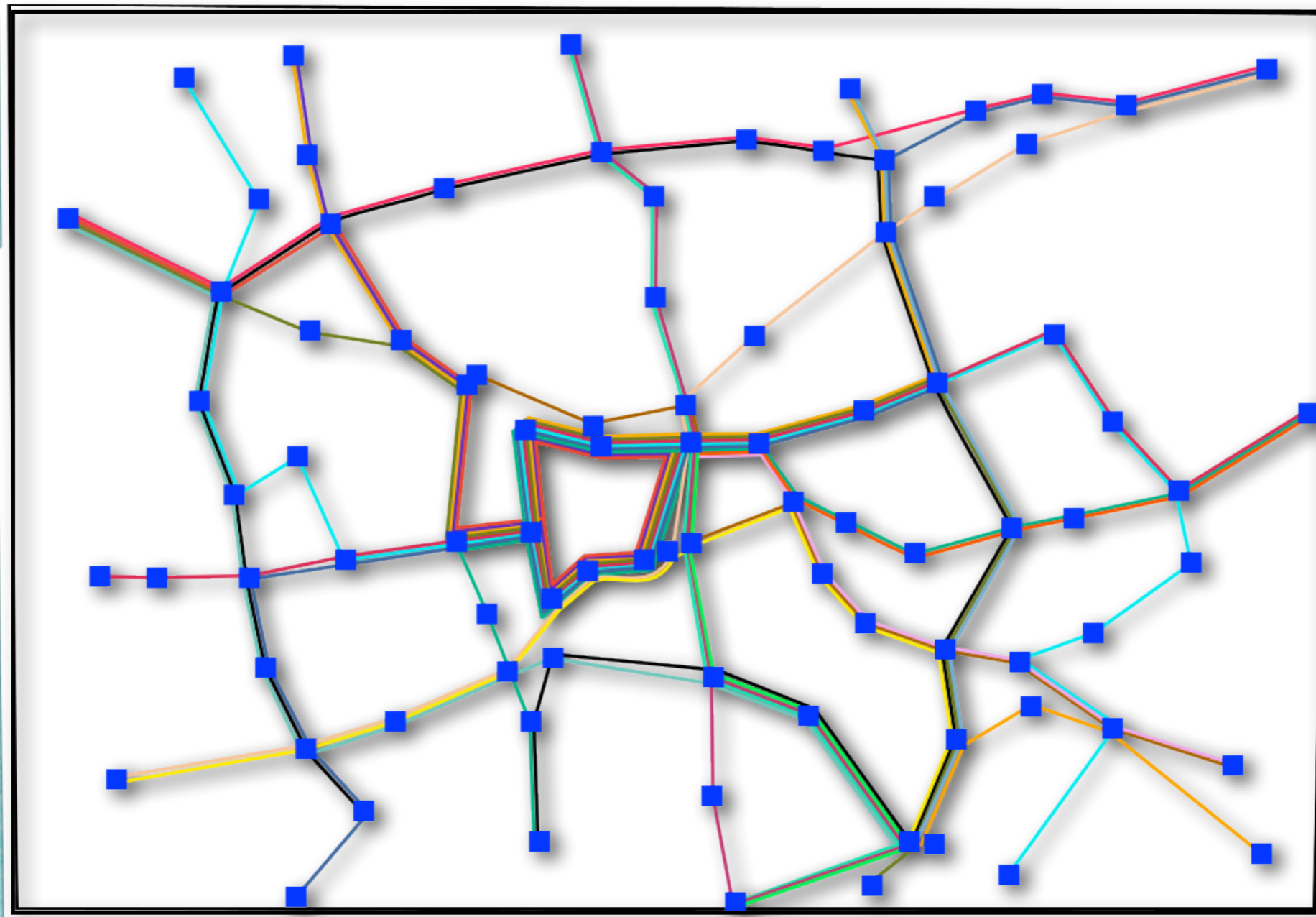
- Solange du noch nicht aufgestanden warst:
  - ▶ Wenn ein oder mehrere direkte Nachbarn aufstehen:
    1. Einen dieser Nachbarn merken
    2. In der nächsten Runde:
      - 2.1. aufstehen
      - 2.2. Zahl merken
    3. In der übernächsten Runde hinsetzen

C. Nach „ANGEKOMMEN!“:

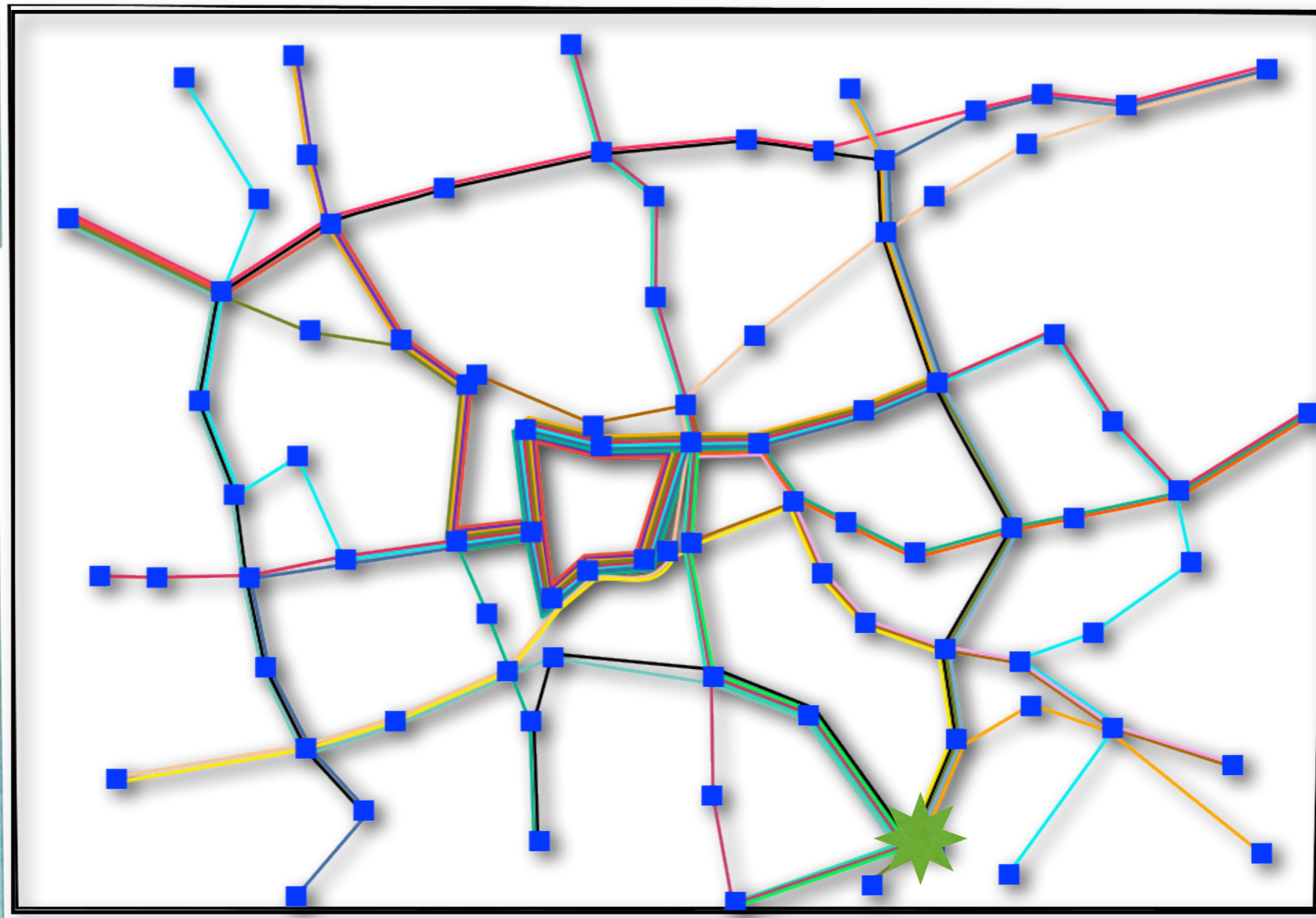
- Auf gemerkten Nachbarn zeigen



# Wellenreiten in Graphen

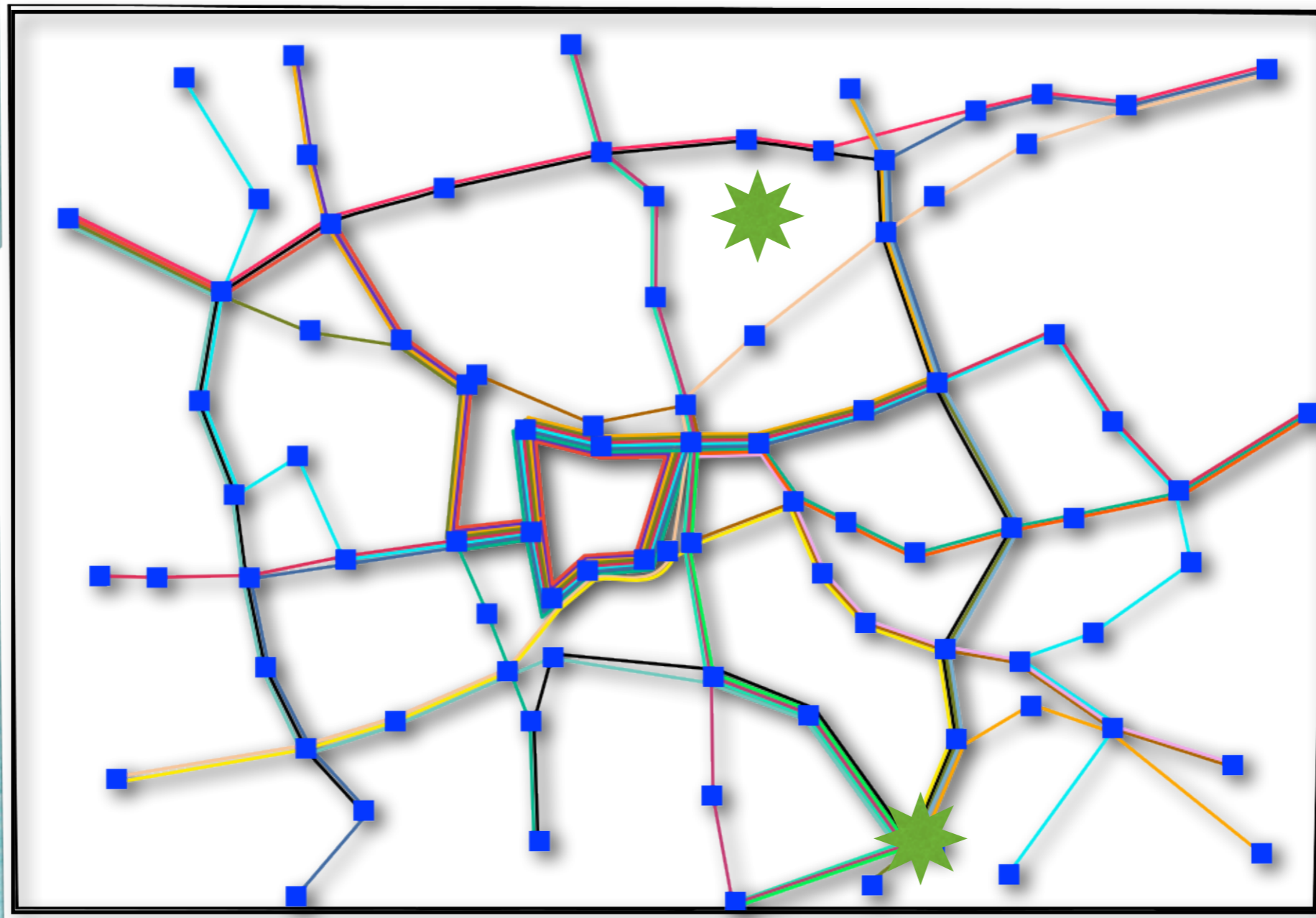


# Wellenreiten in Graphen

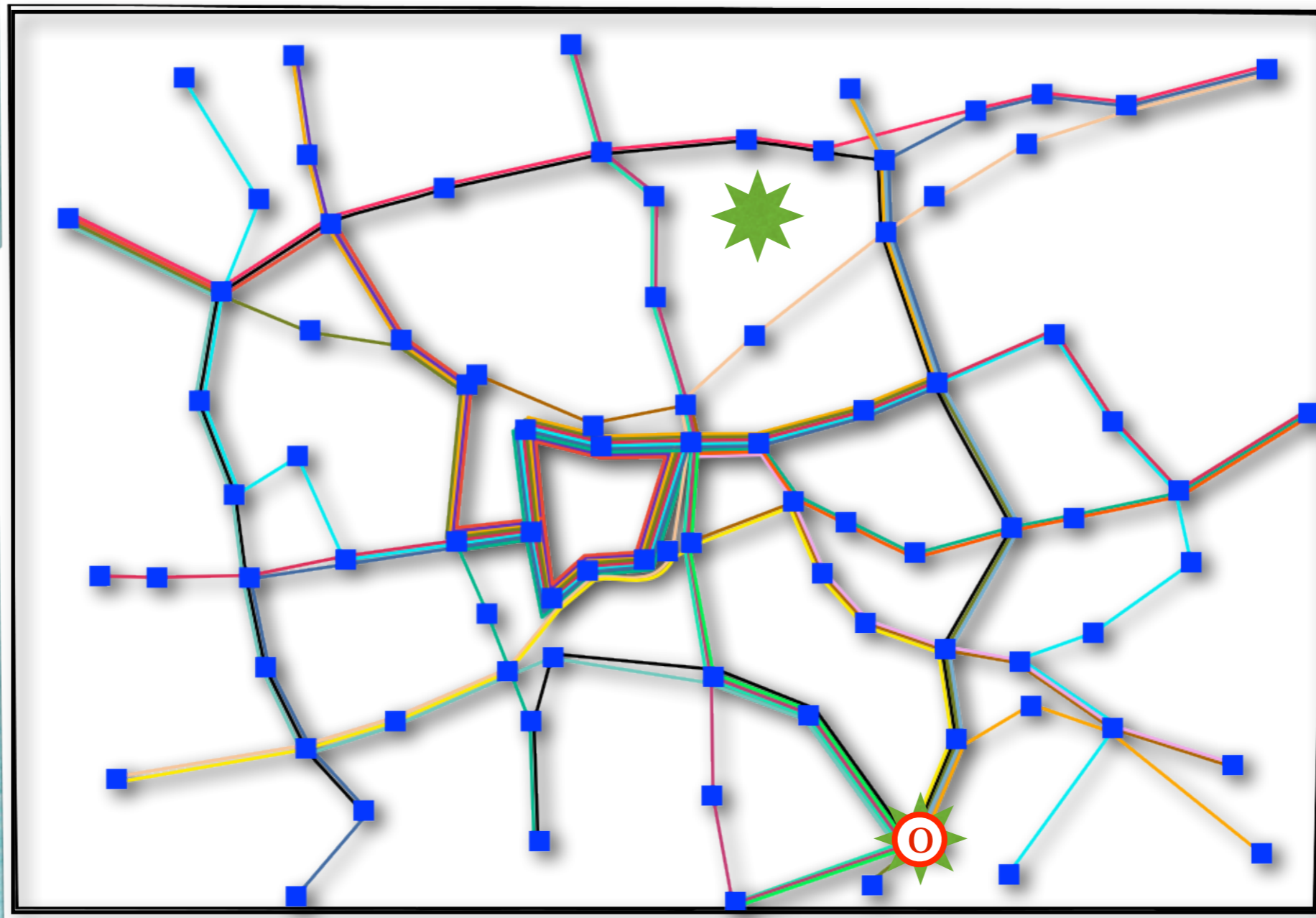




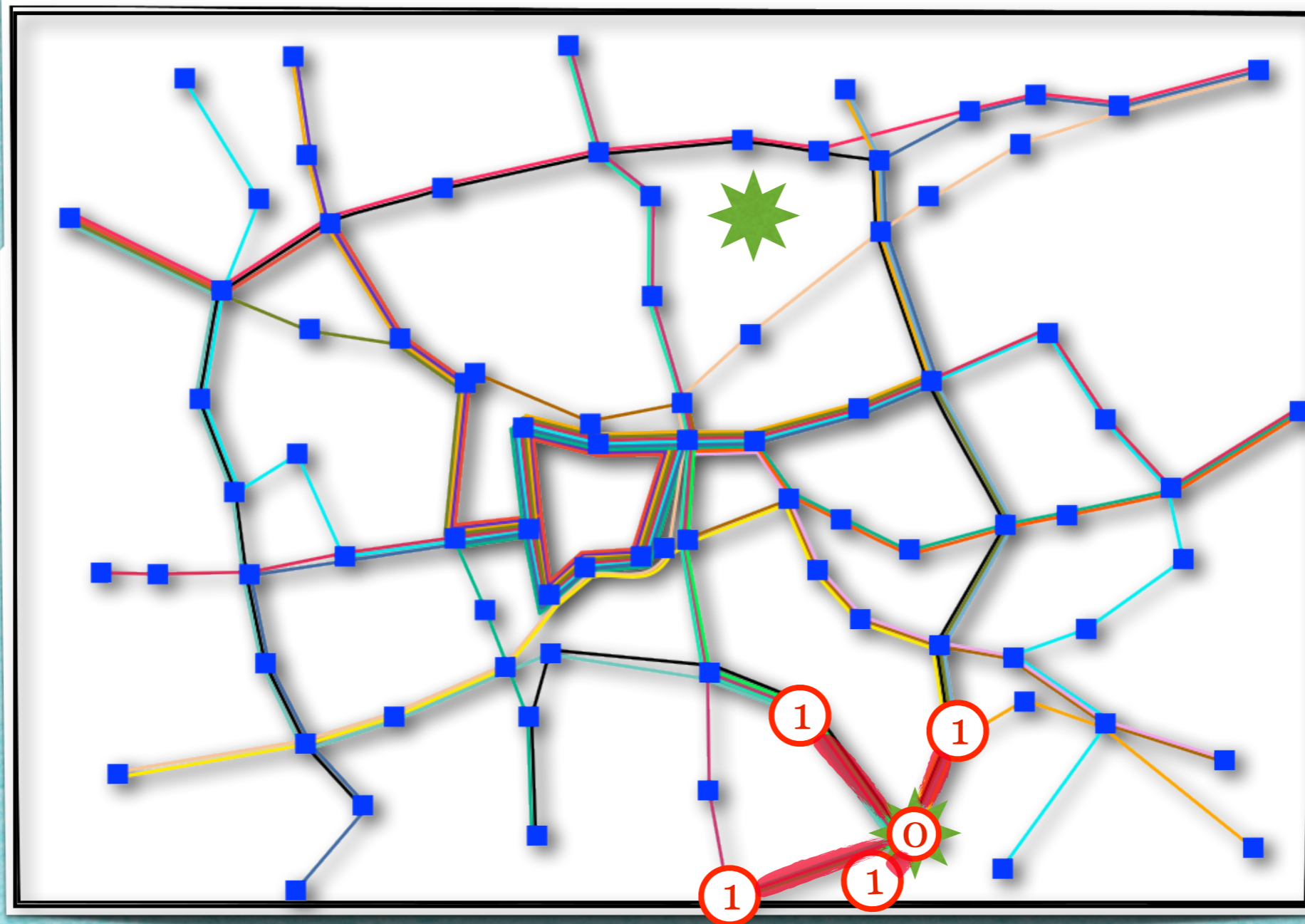
# Wellenreiten in Graphen



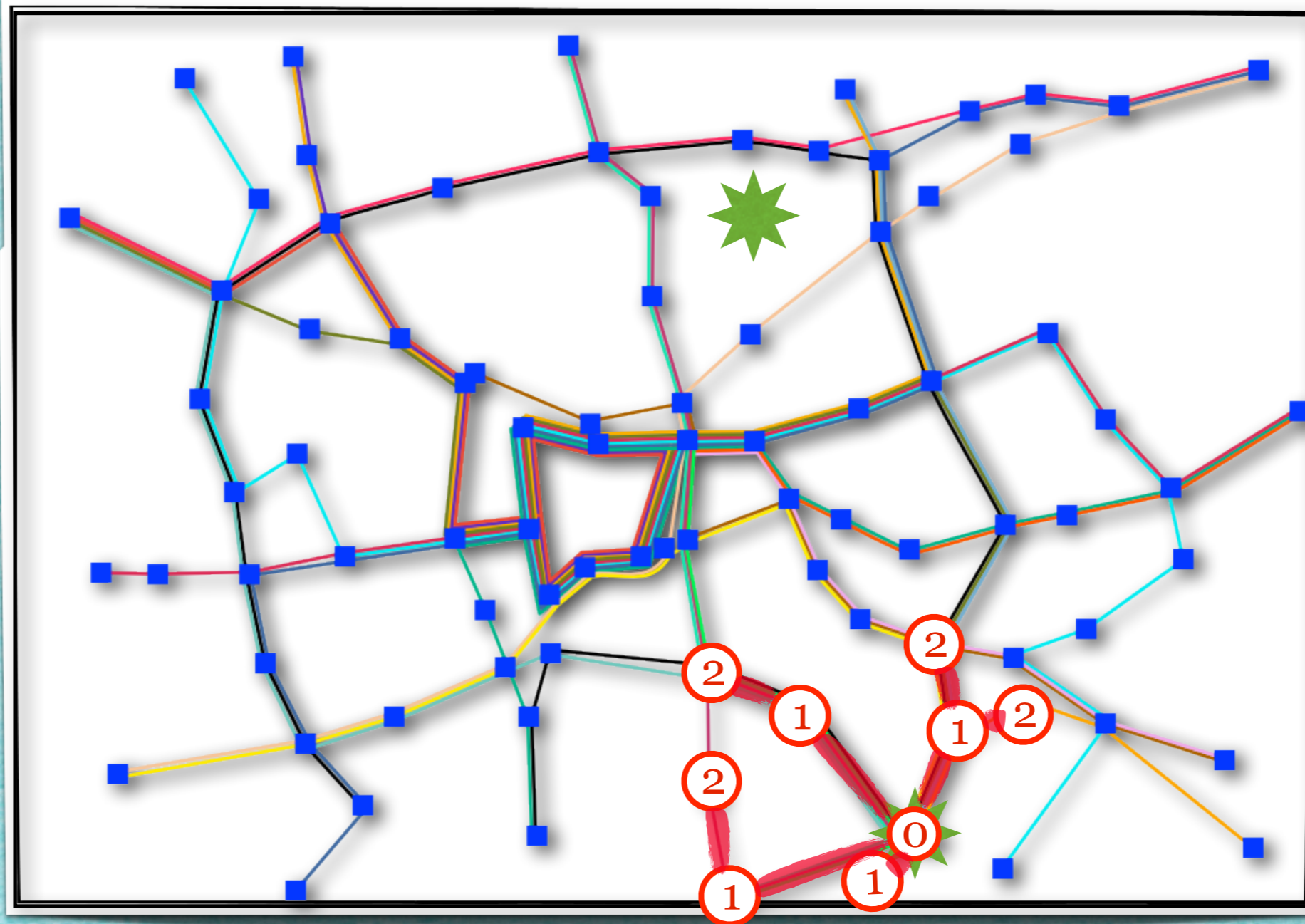
# Wellenreiten in Graphen



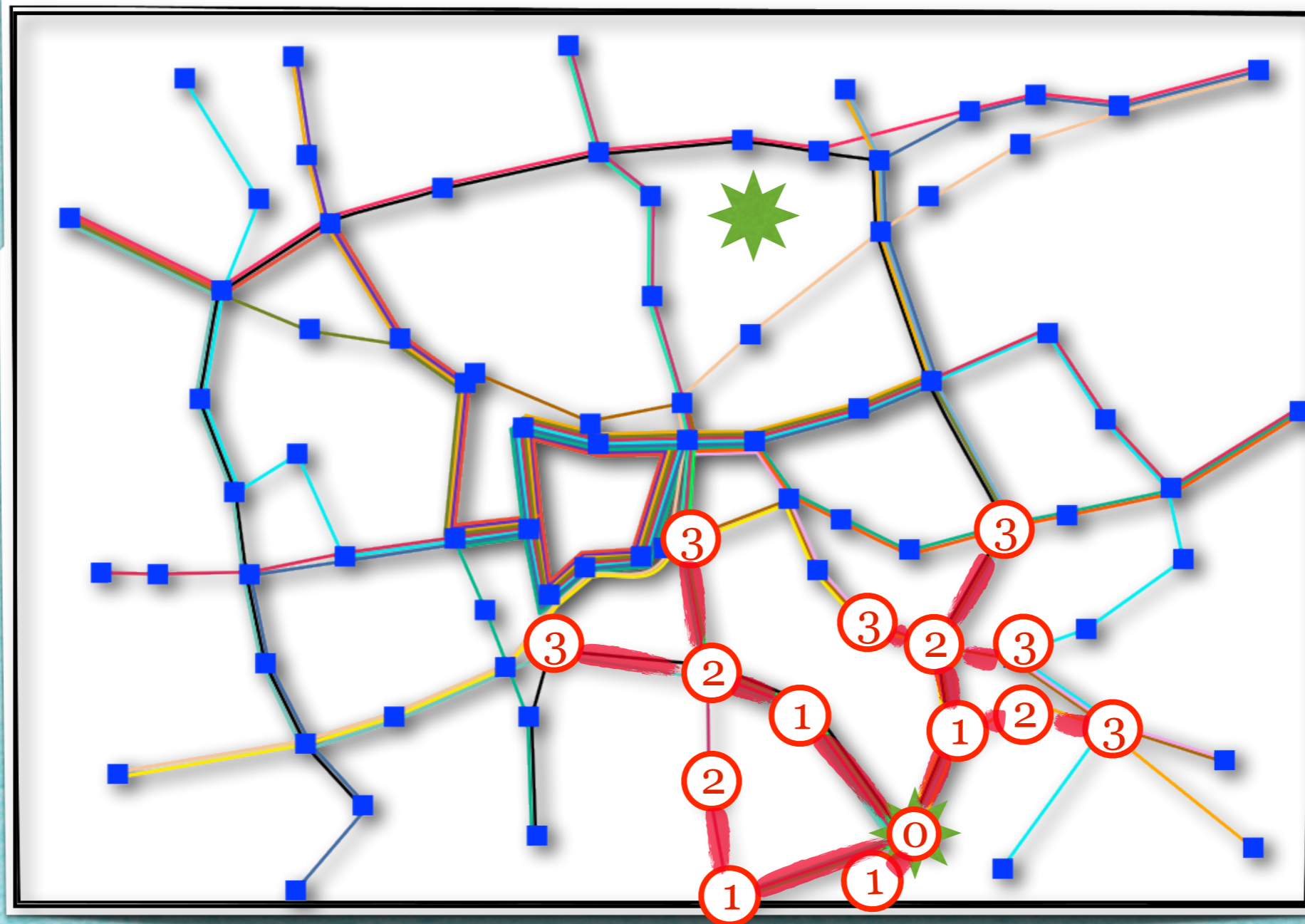
# Wellenreiten in Graphen



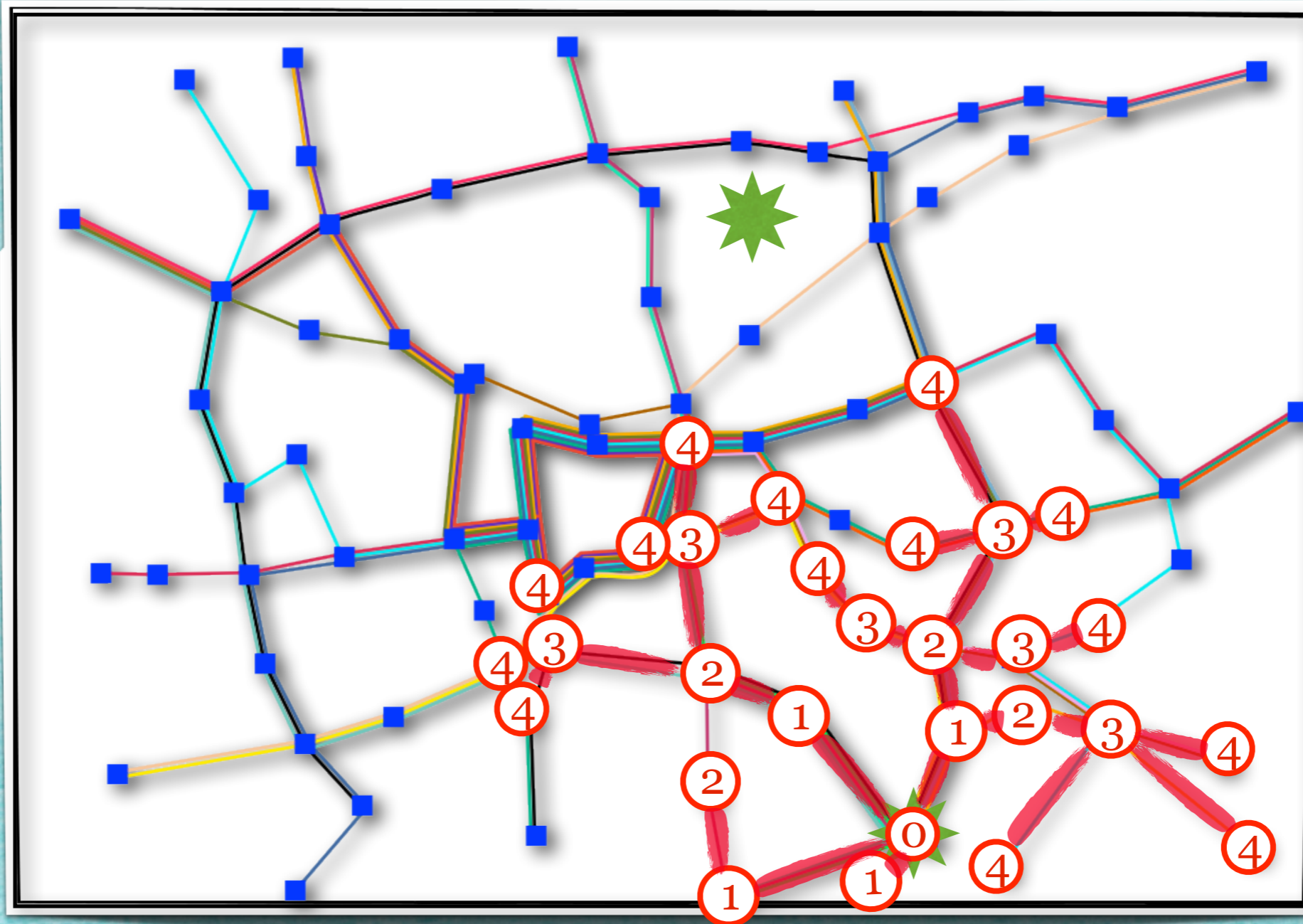
# Wellenreiten in Graphen



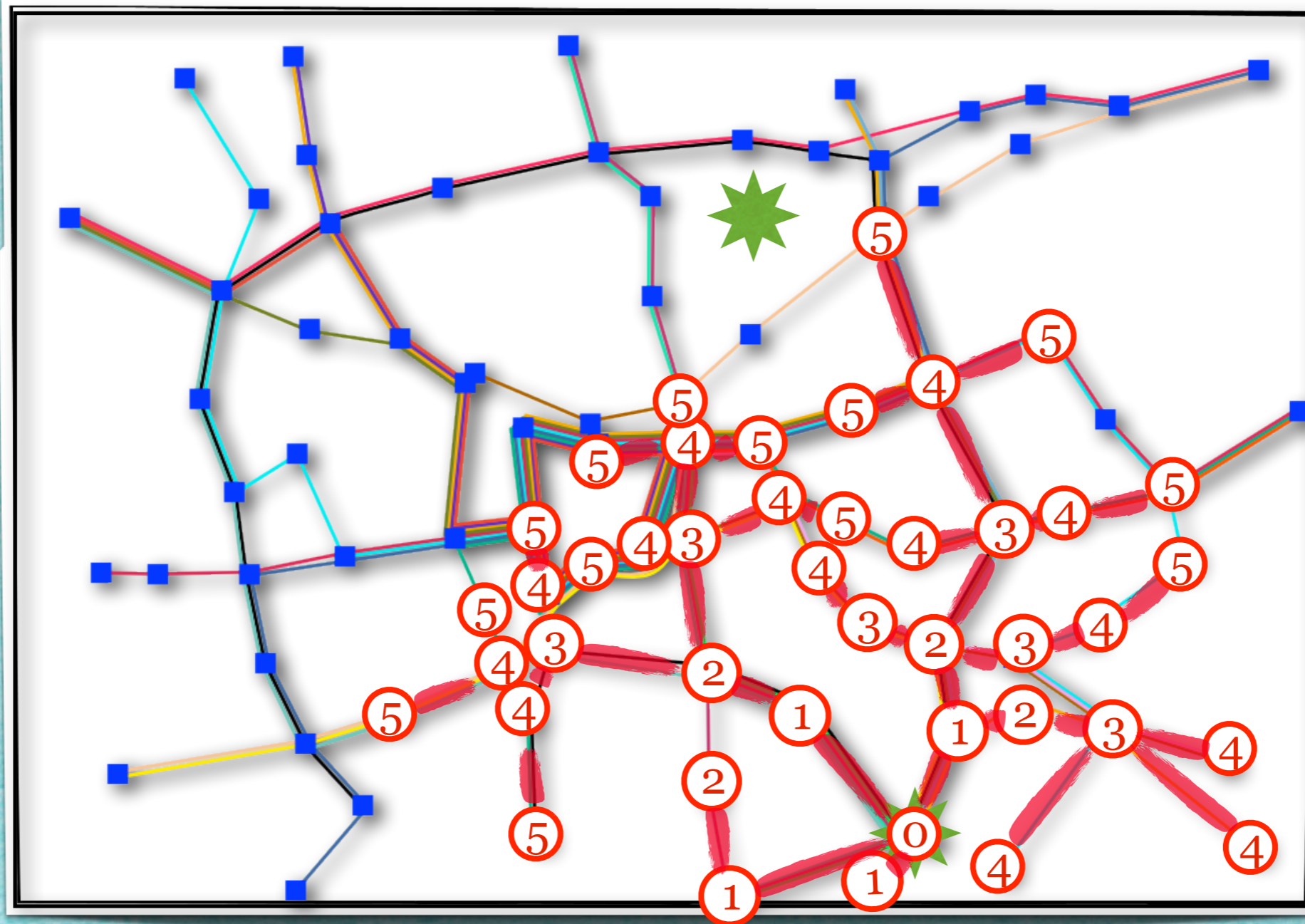
# Wellenreiten in Graphen



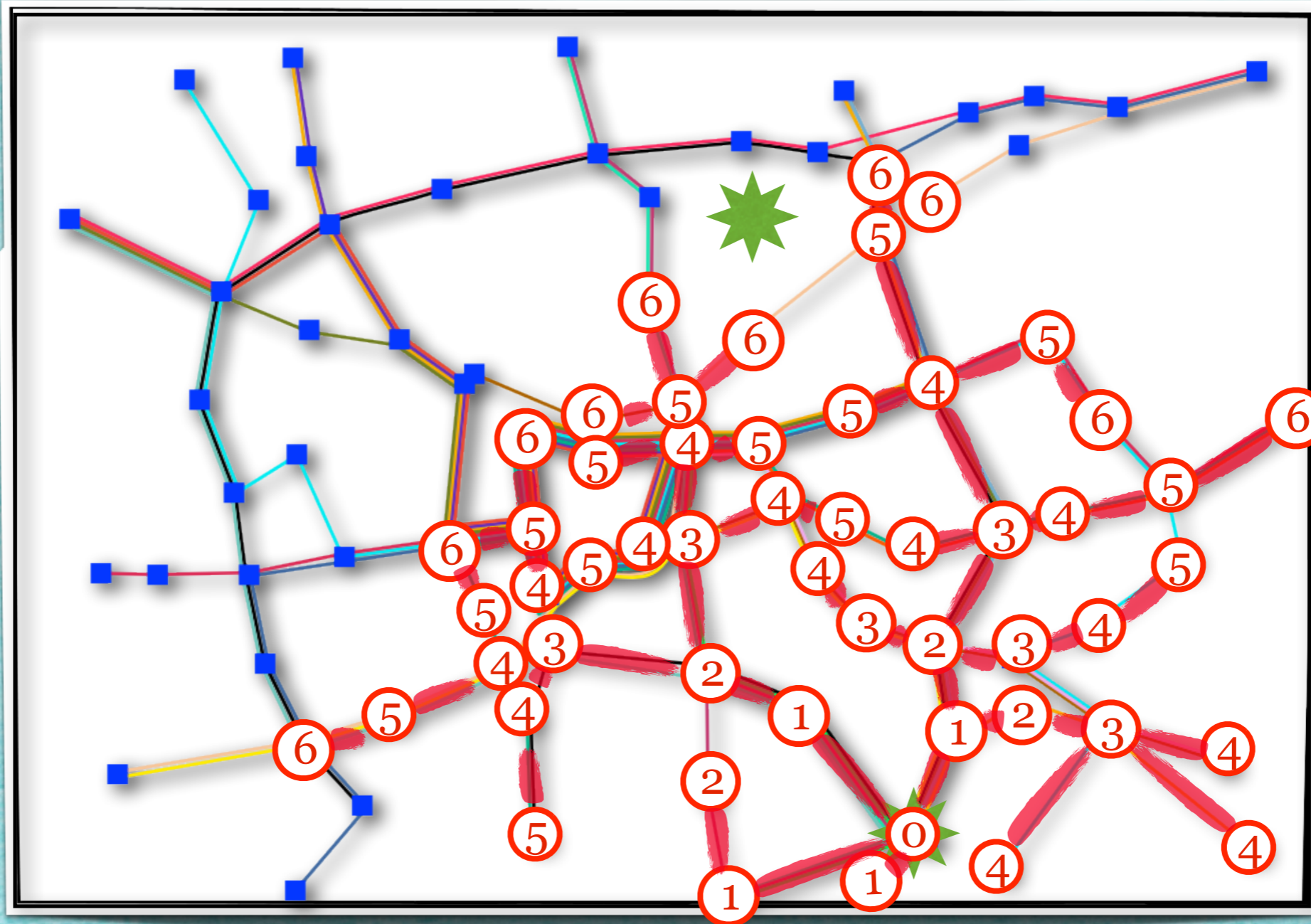
# Wellenreiten in Graphen



# Wellenreiten in Graphen

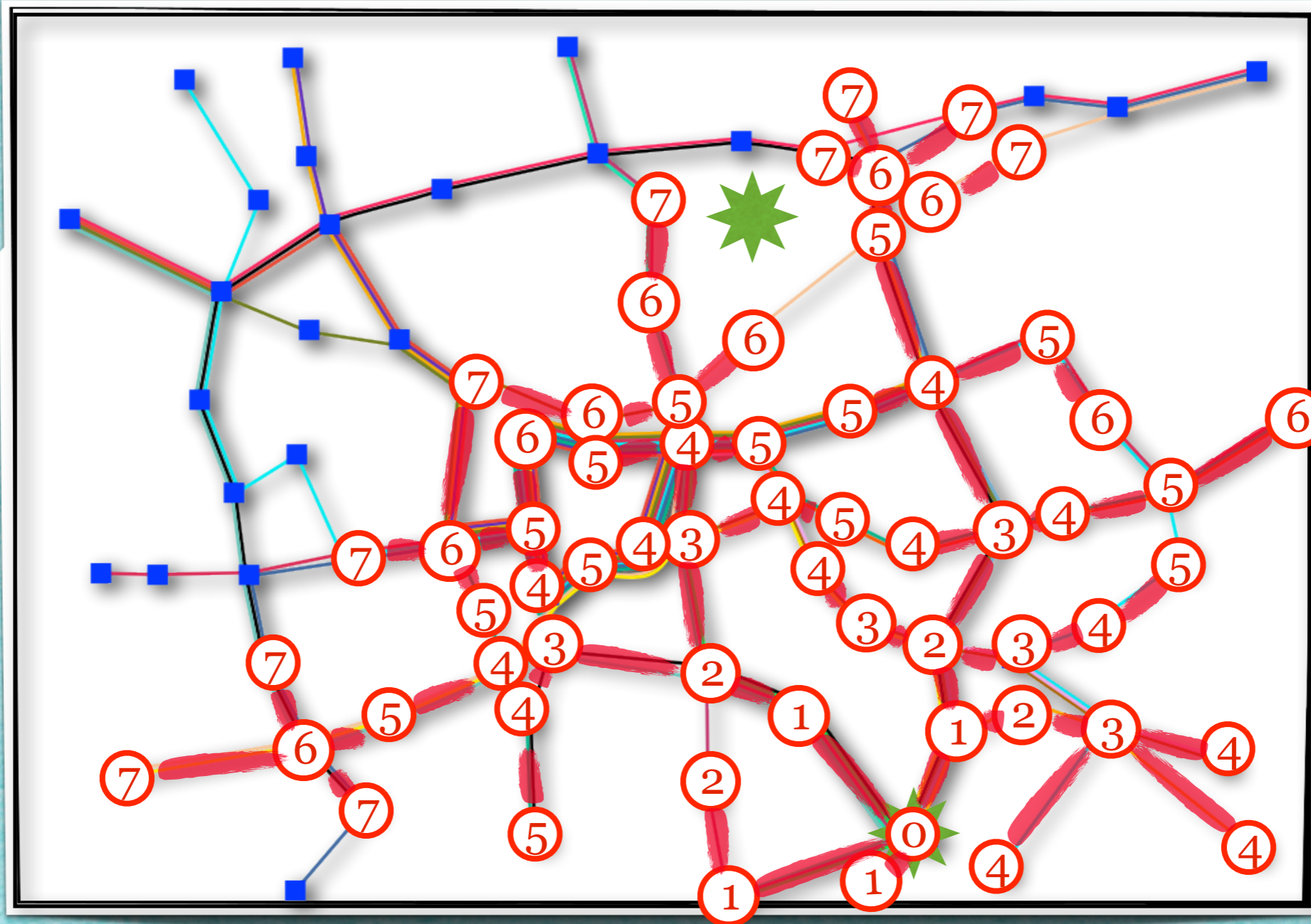


# Wellenreiten in Graphen

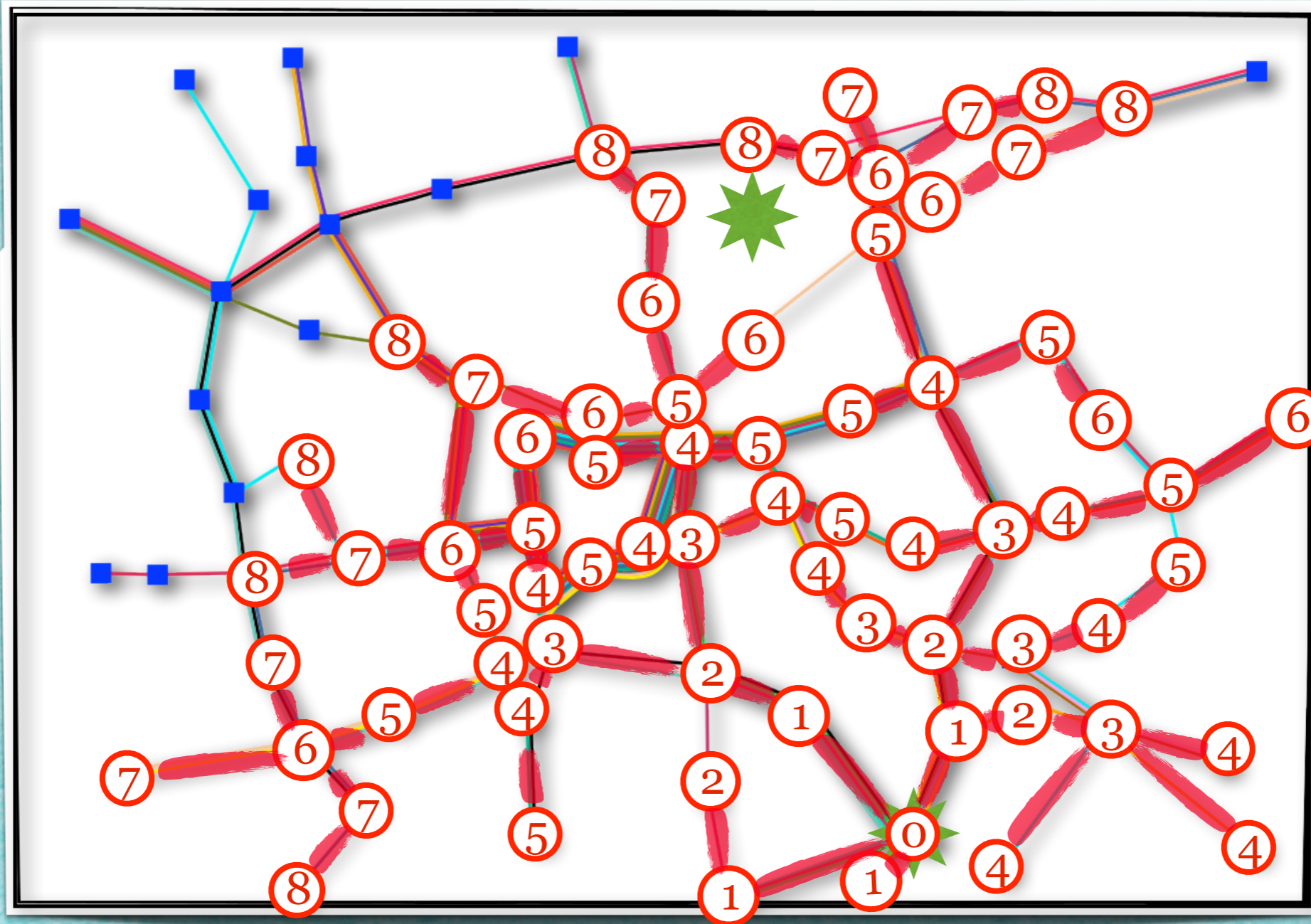




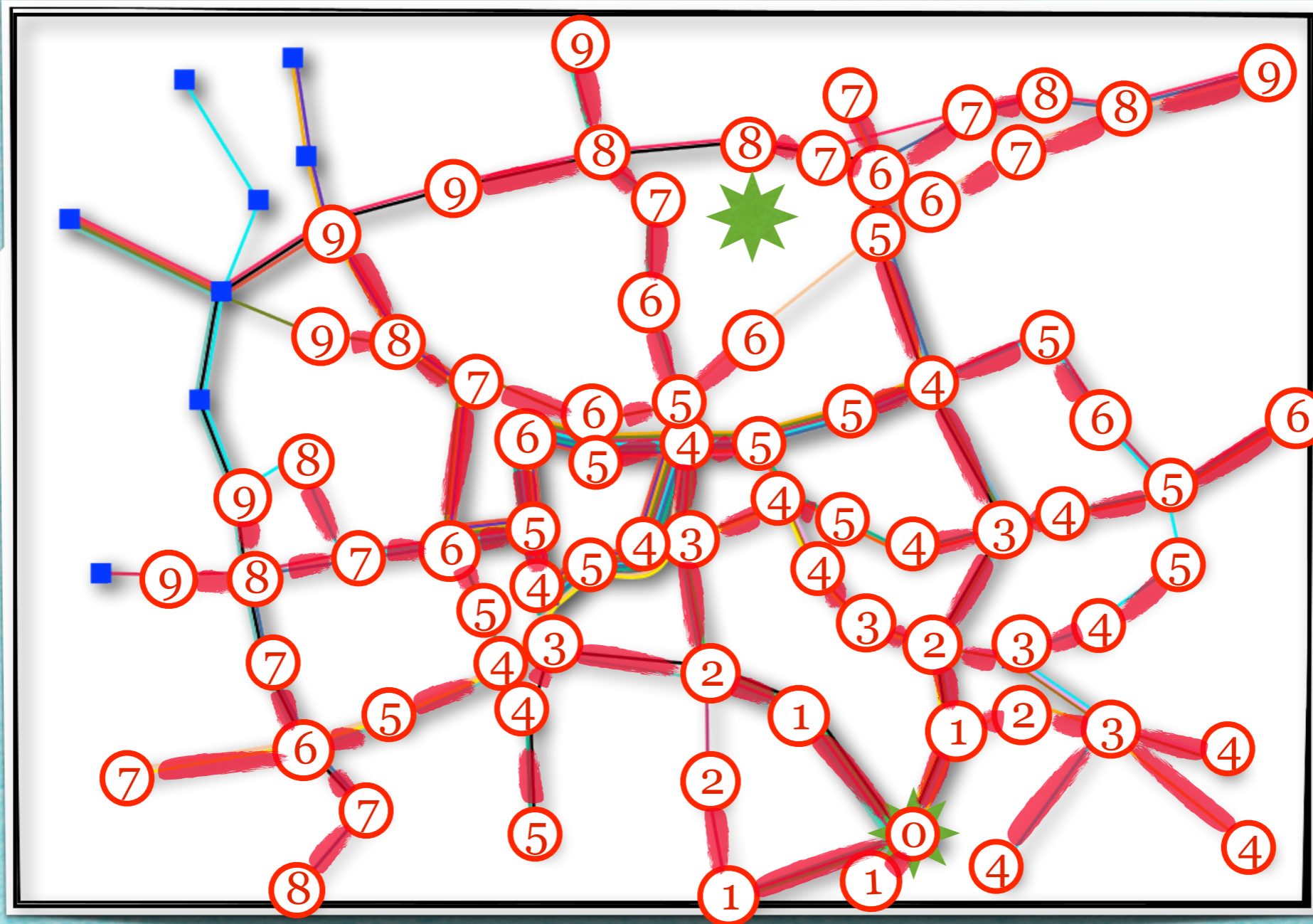
# Wellenreiten in Graphen



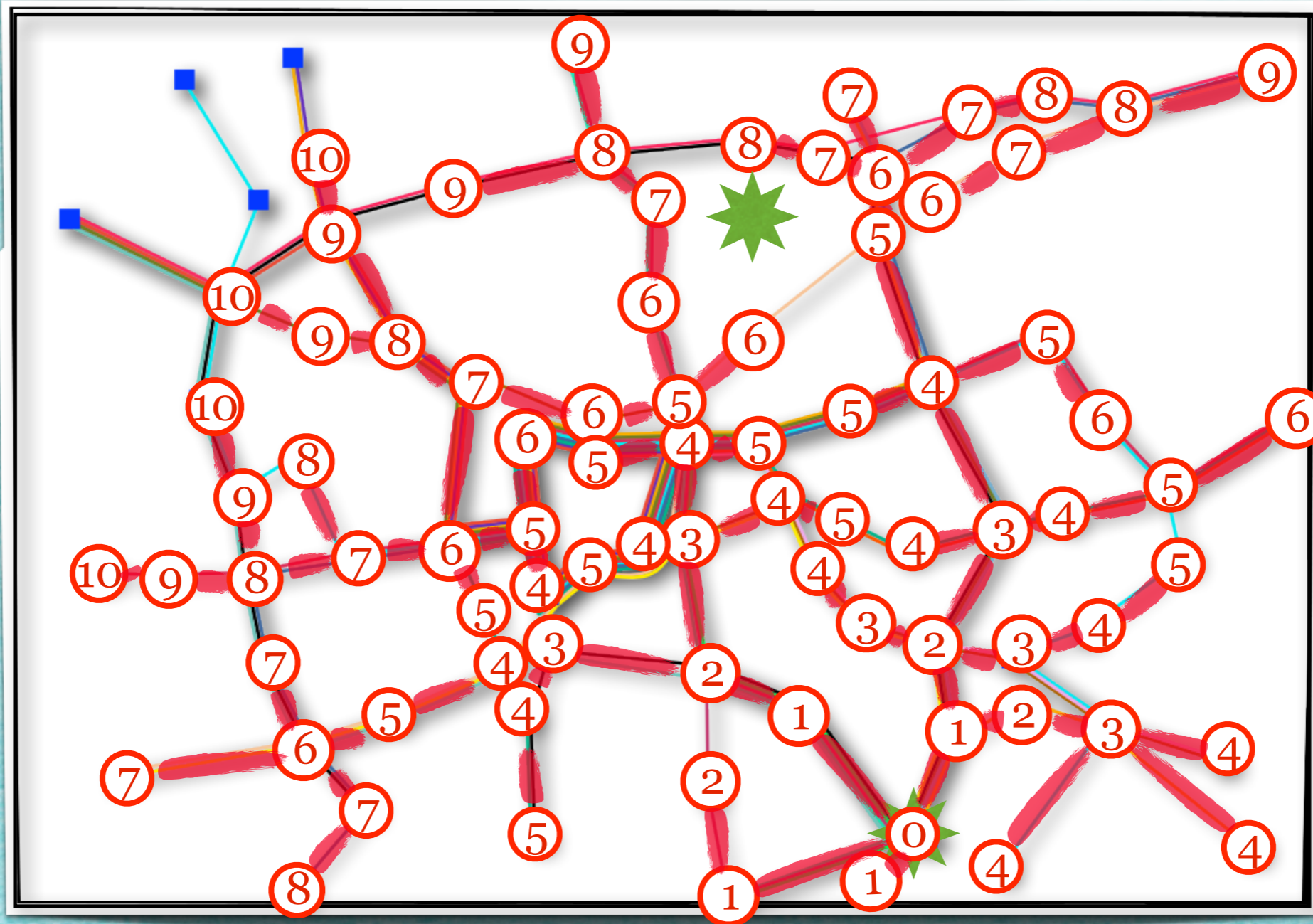
# Wellenreiten in Graphen



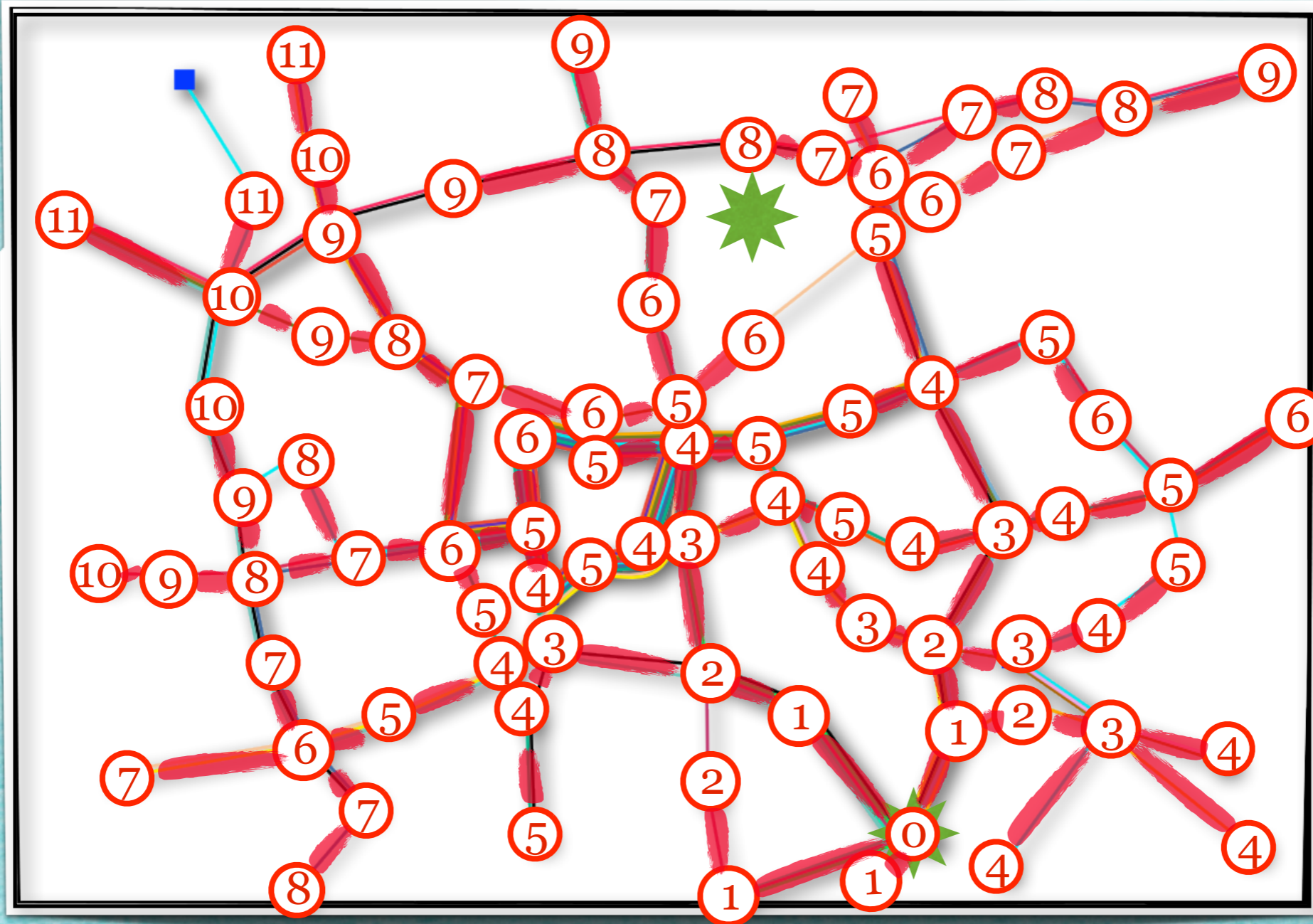
# Wellenreiten in Graphen



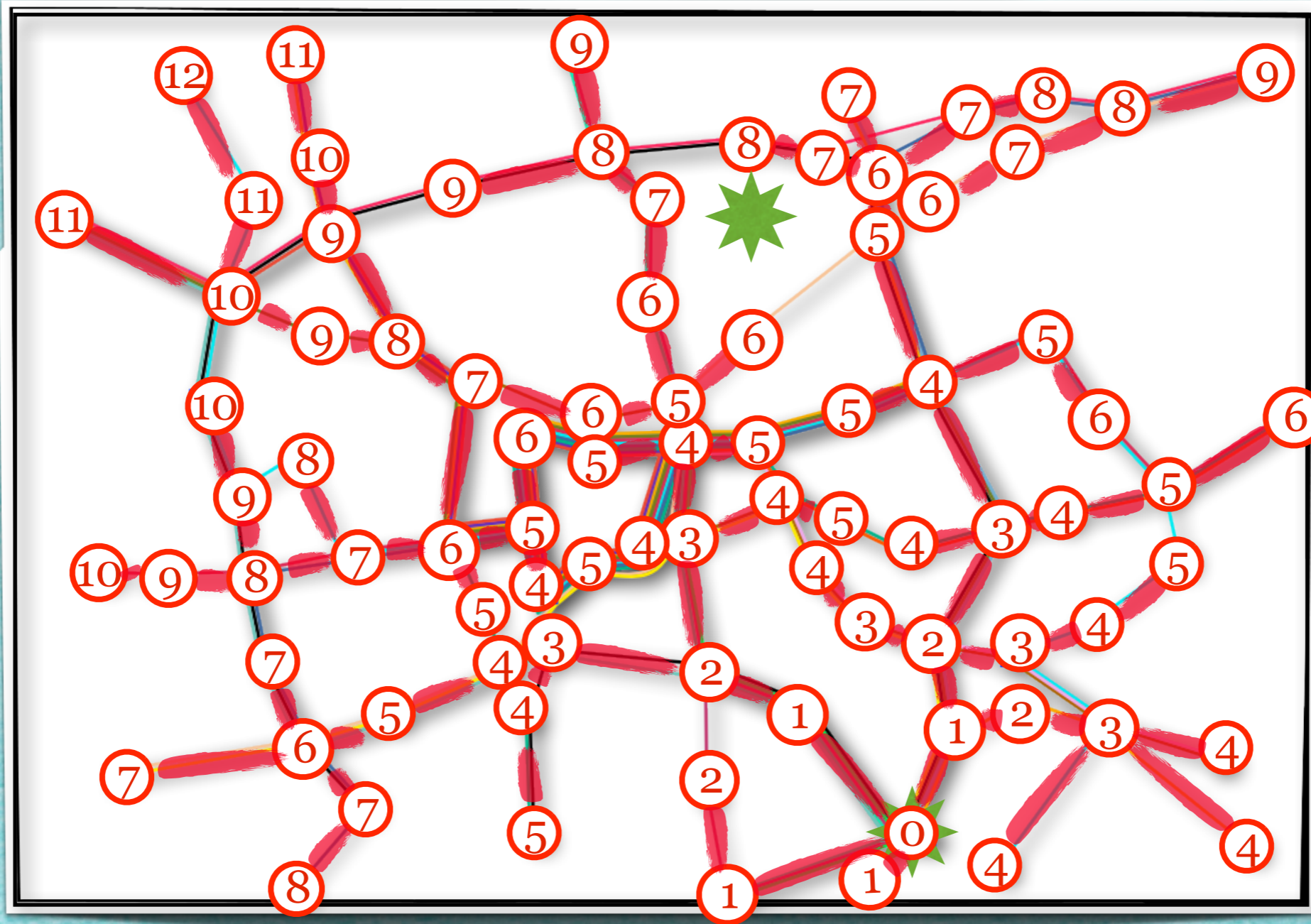
# Wellenreiten in Graphen



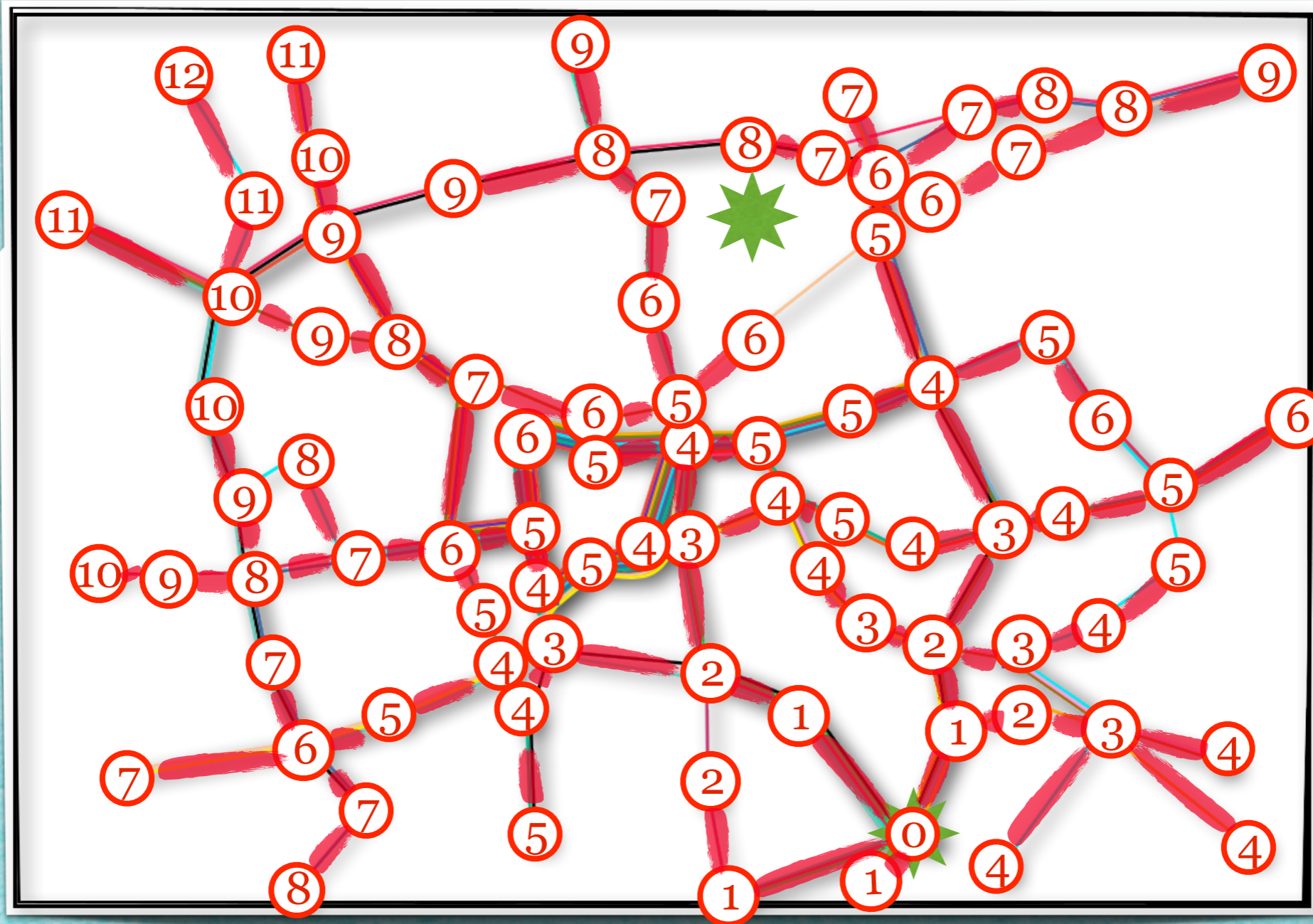
# Wellenreiten in Graphen



# Wellenreiten in Graphen

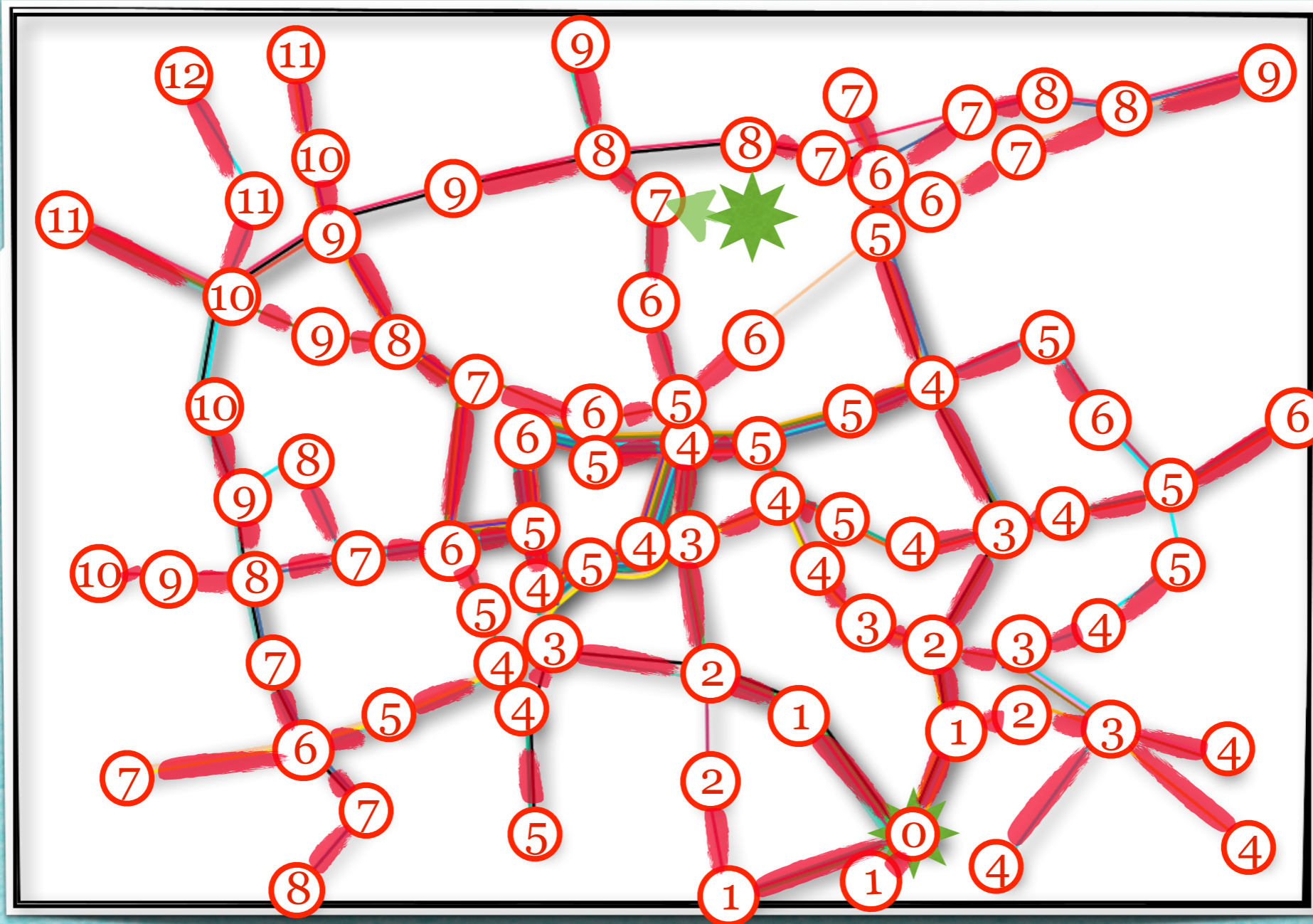


# Wellenreiten in Graphen



Breitensuche

# Wellenreiten in Graphen



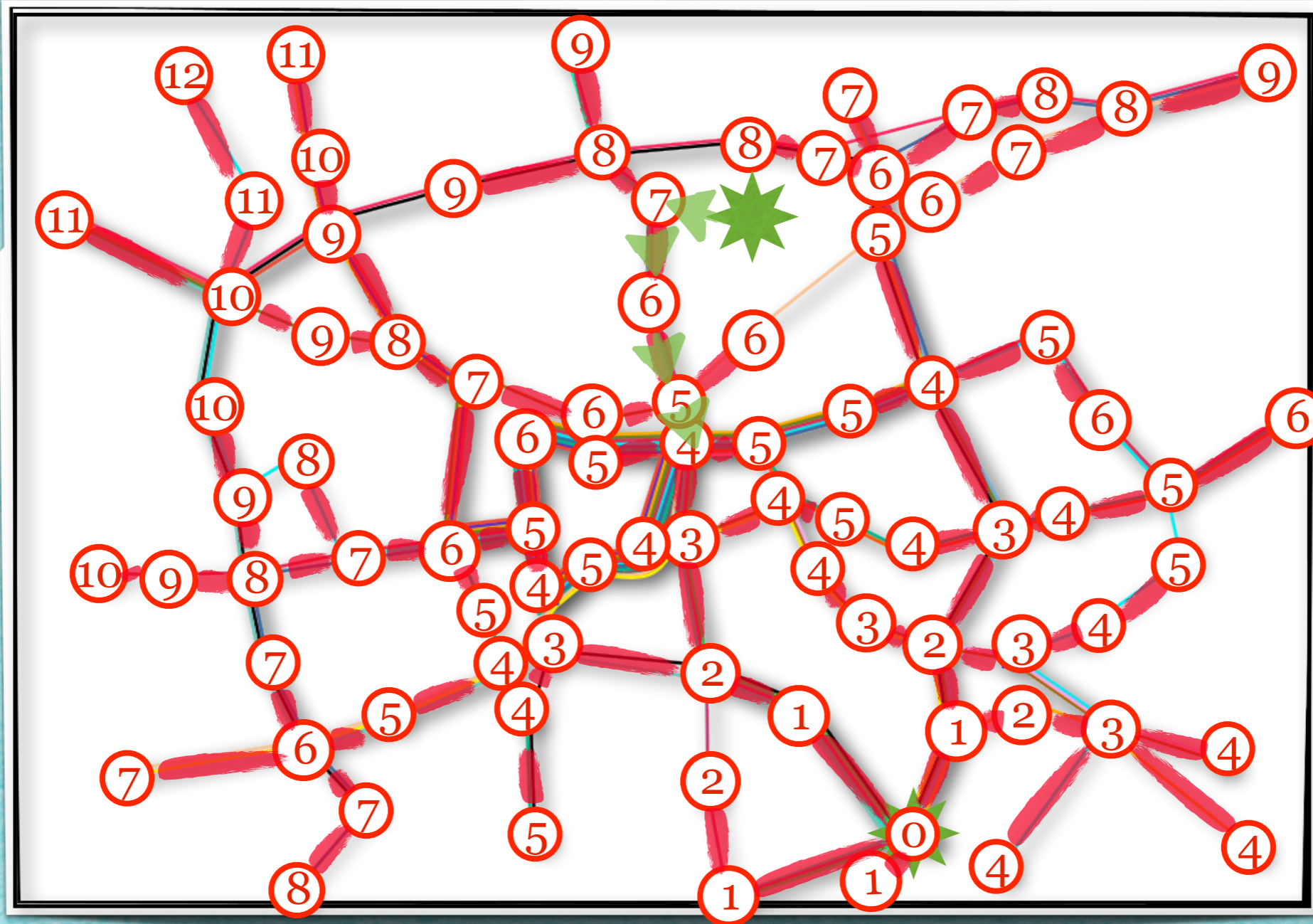
Breitensuche





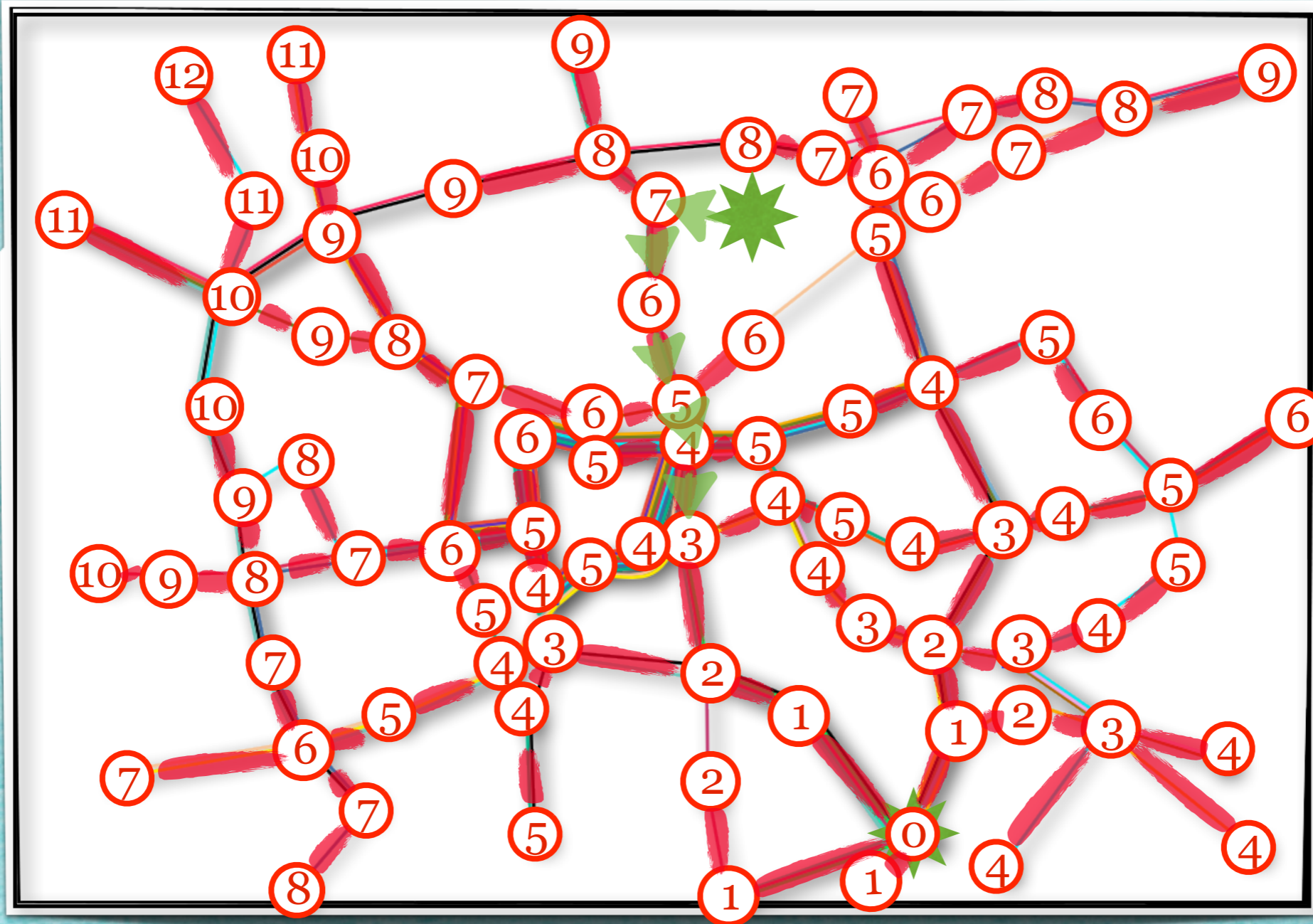


# Wellenreiten in Graphen



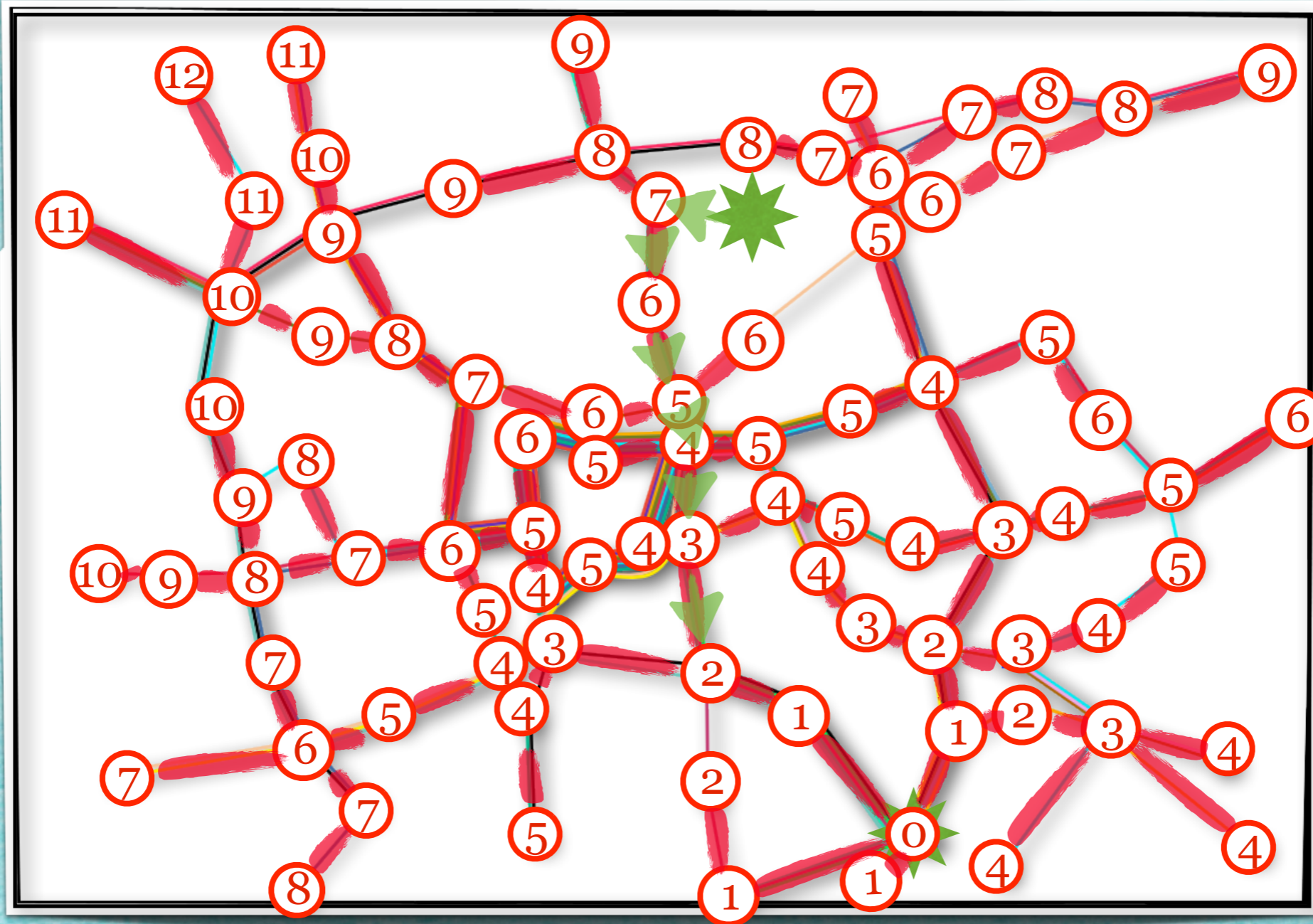
Breitensuche

# Wellenreiten in Graphen



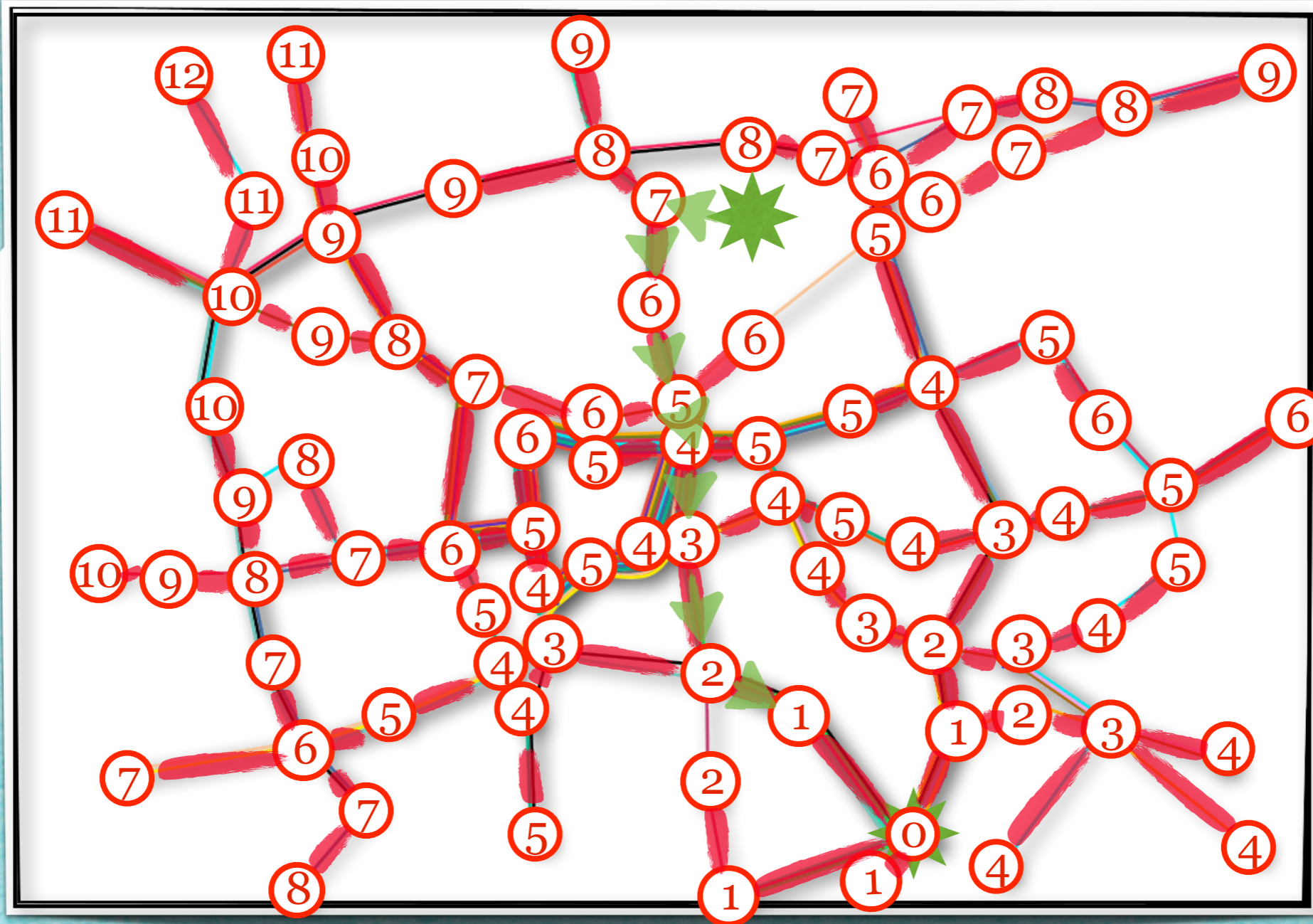
Breitensuche

# Wellenreiten in Graphen



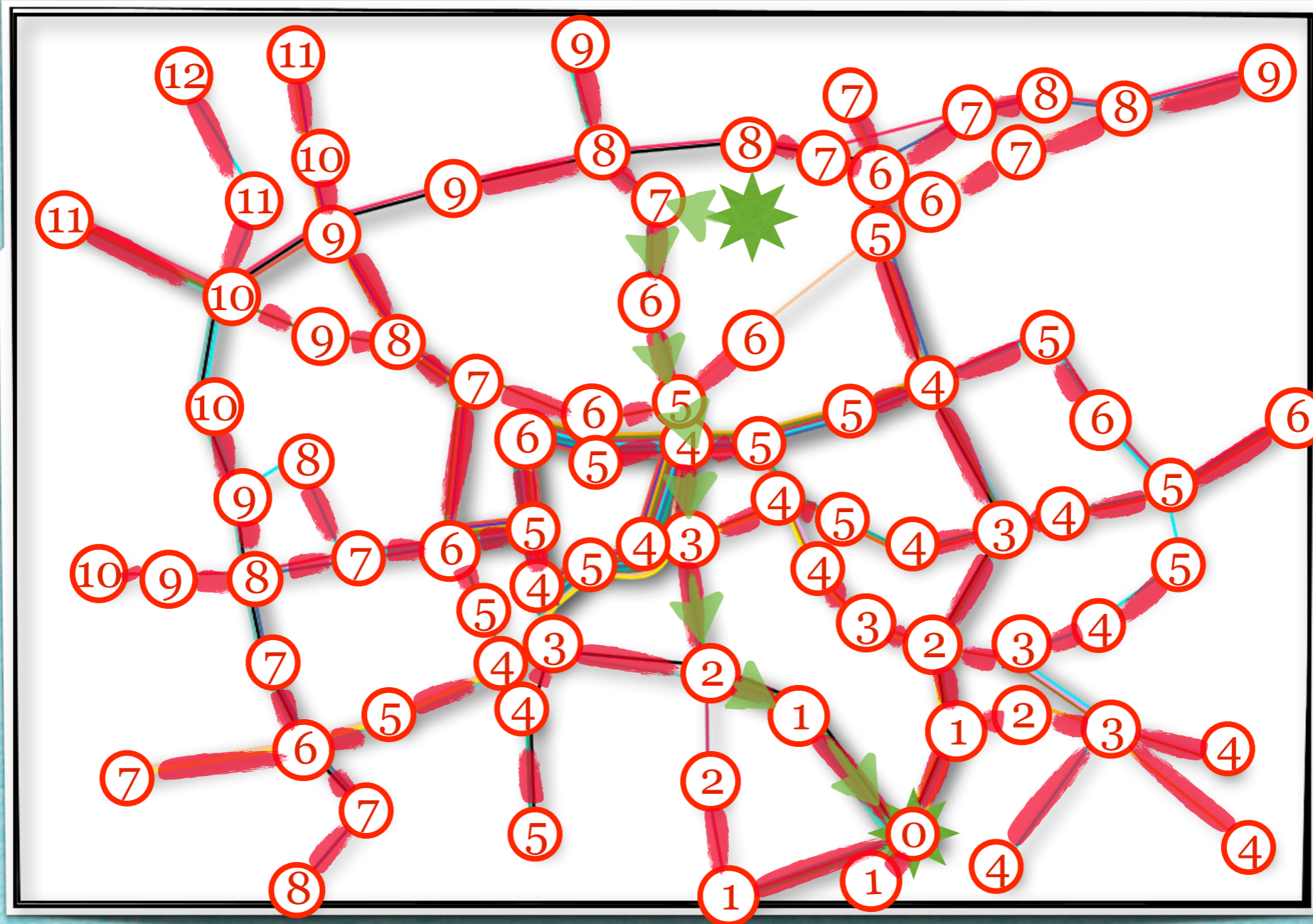
Breitensuche

# Wellenreiten in Graphen



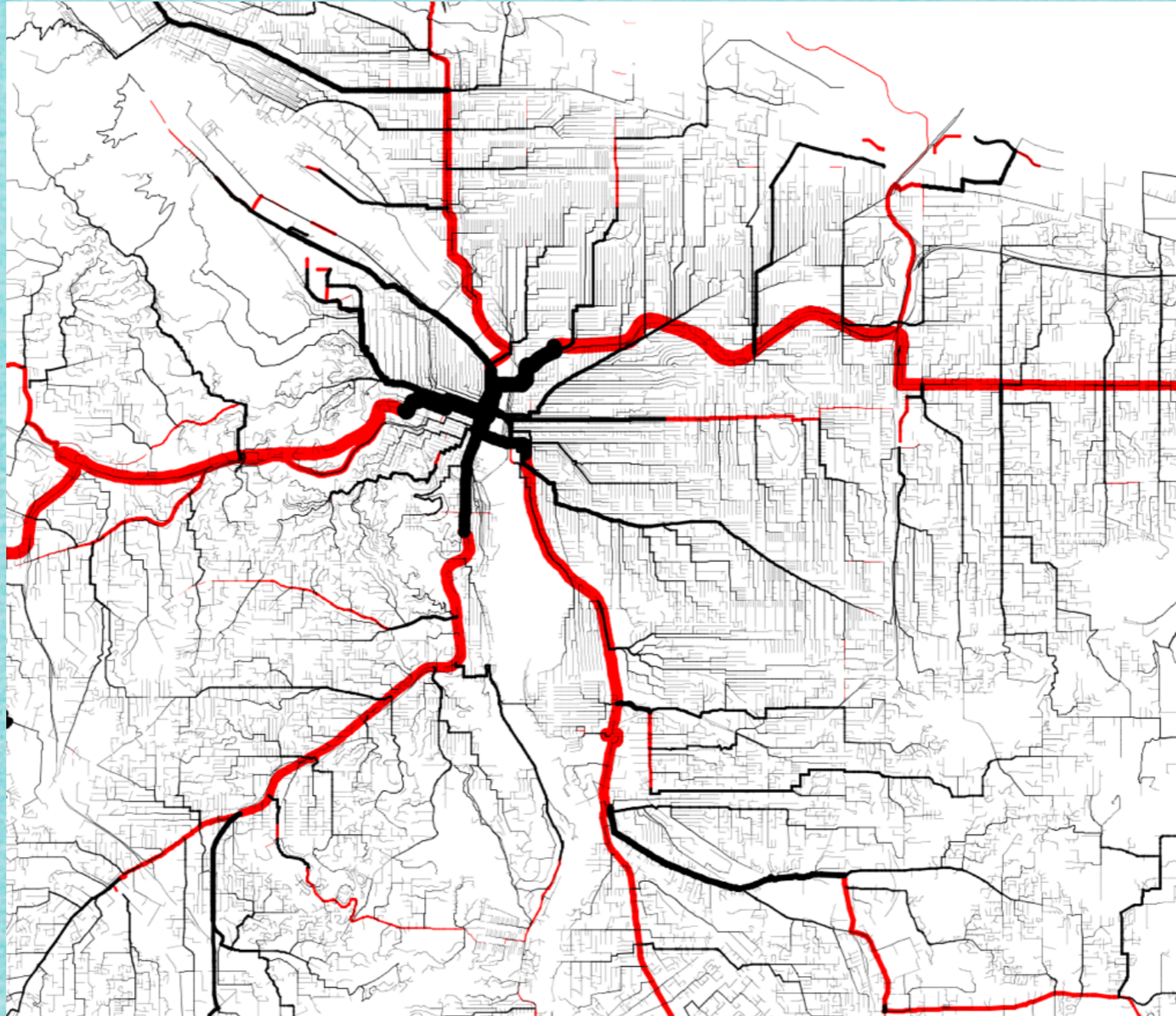
Breitensuche

# Wellenreiten in Graphen



Breitensuche

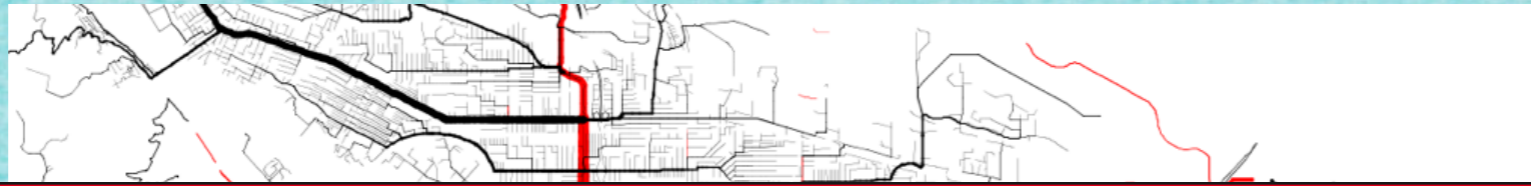
**Breitensuche liefert kürzeste Wege von einer Quelle aus**



**Viele "Sammler"**

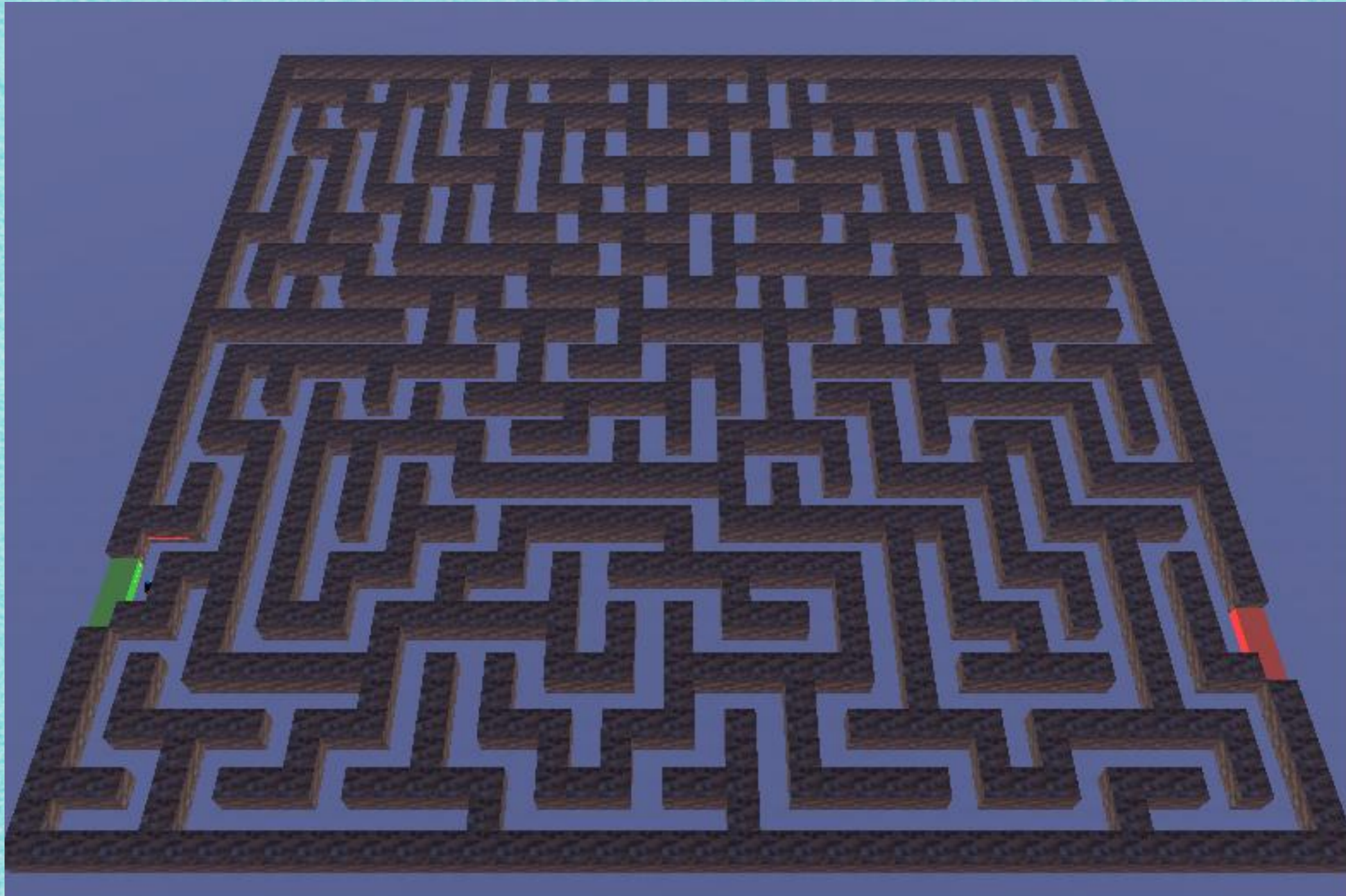


Breitensuche liefert kürzeste Wege von einer Quelle aus



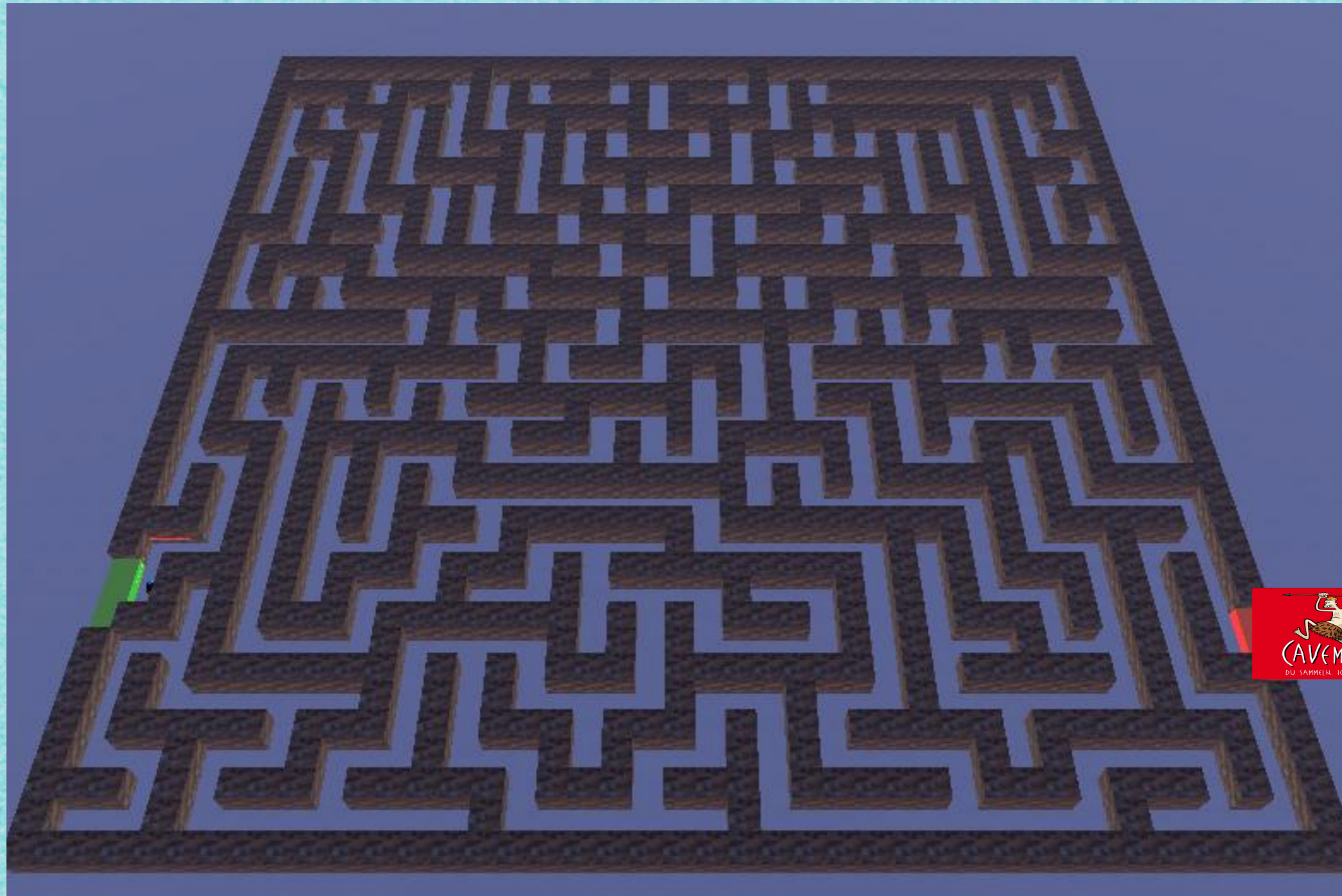
Viele "Sammler"

Tiefensuche findet lokal einen Weg durch ein Labyrinth



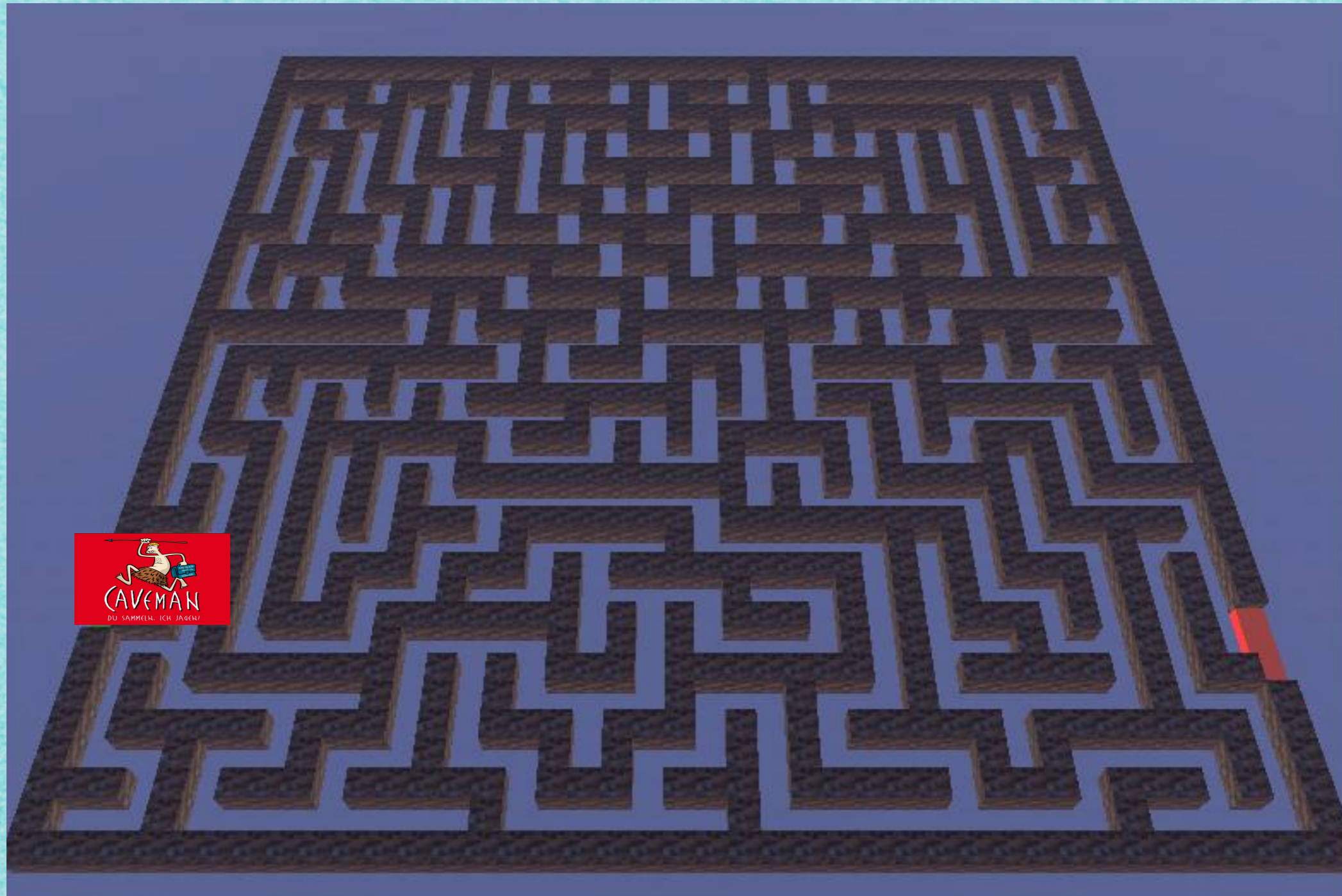
Ein "Jäger"

Tiefensuche findet lokal einen Weg durch ein Labyrinth



Ein "Jäger"

Tiefensuche findet lokal einen Weg durch ein Labyrinth



Ein "Jäger"



Zum dritten Mal ein Paar

## Bettina und Christian Wulff zeigen sich erstmals wieder gemeinsam

Der ehemalige Bundespräsident Christian Wulff und seine Frau Bettina wagen zum dritten Mal einen Neustart ihrer Beziehung. Nun ist das Paar gemeinsam bei einem Presse-Event aufgetreten.

22.06.2021, 19.17 Uhr





"I told you we should have asked for directions."

WENN NAVIGATIONSSYSTEME WIRKLICH WEIBLICH WÄREN...

NA GUT, DANN KANN ICH HALT KEINE KARTEN LESEN!

ABER WER WOLLTE DENN NICHT ANHALTEN UND NACH DEM WEG FRAGEN?!



MYCARTOONS.DE

WHY DID MOSES WANDER IN THE DESERT FOR FORTY YEARS?

BECAUSE EVEN BACK THEN A MAN WOULDN'T STOP AND ASK FOR DIRECTIONS!



NOW WILL YOU ASK FOR DIRECTIONS?



"Alright then, I'll admit it... we're lost!"

*Mehr demnächst!*

*s.fekete@tu-bs.de*