



*Kapitel 3.4:*  
*Wartenschlange und Stapel*  
*Algorithmen und Datenstrukturen*  
*WS 2018/19*

**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

# A & D

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

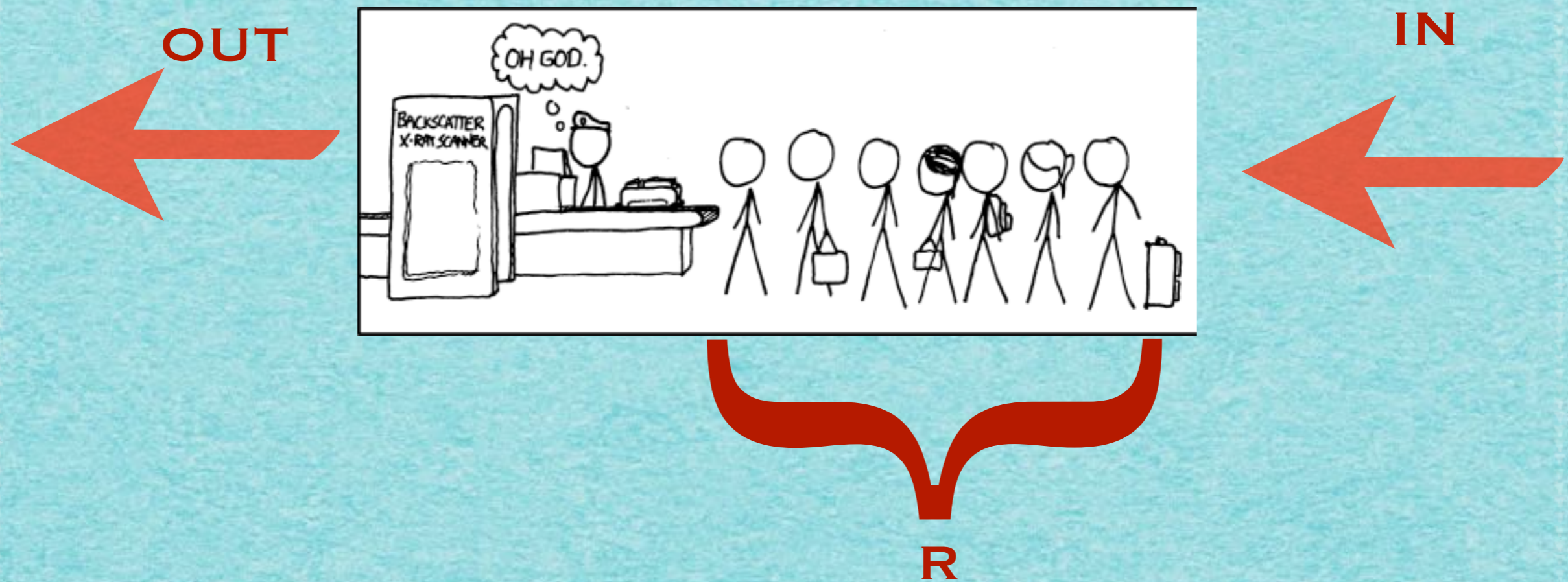
CAN YOU MAKE IT  
WITHOUT KILLING  
YOURSELF?



**Algorithmus**

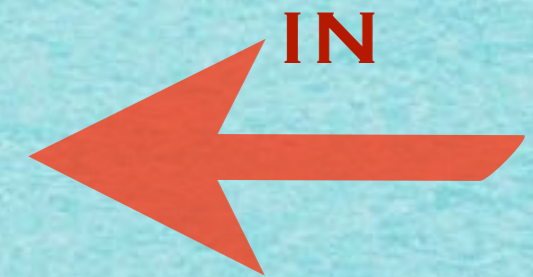
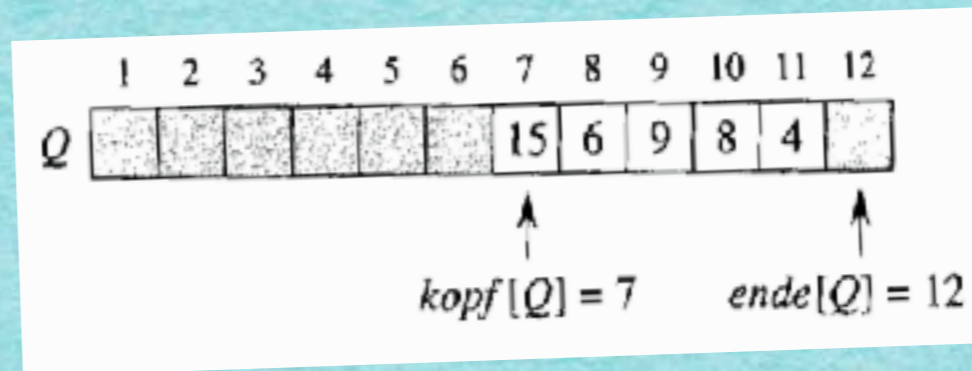
**DATENSTRUKTUR**

# DATENSTRUKTUR I

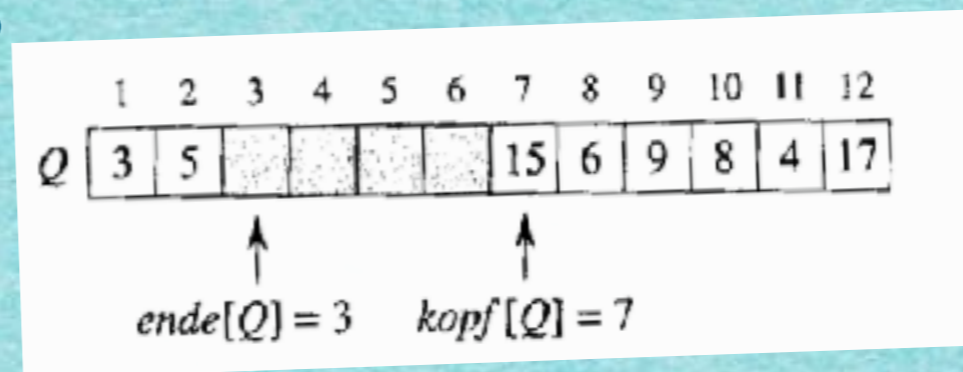


**WARTESCHLANGE: FIRST IN - FIRST OUT**

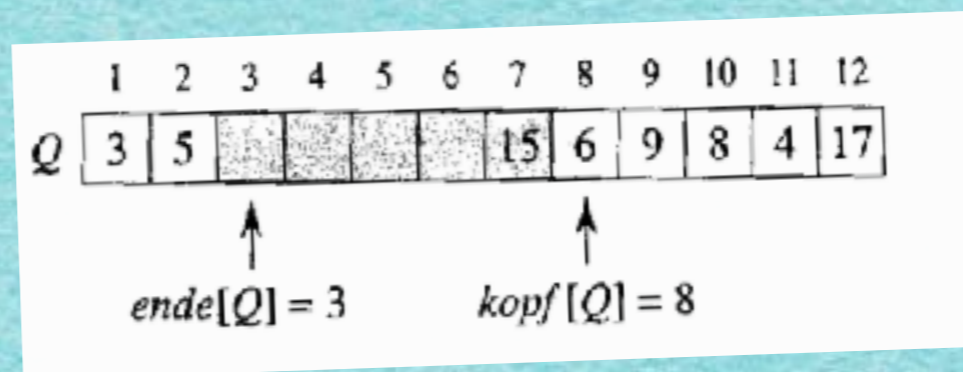
# WARTESCHLANGE AUF ARRAY UMGESETZT



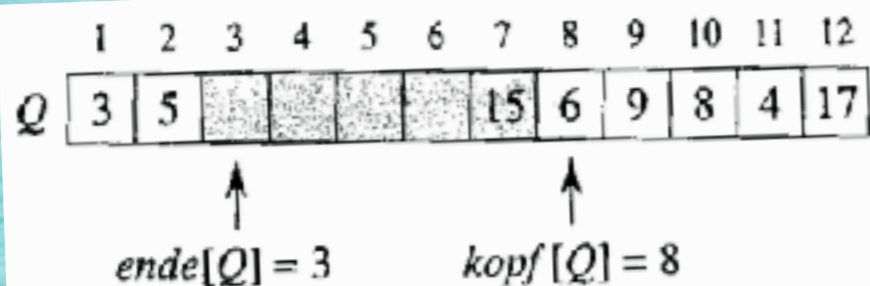
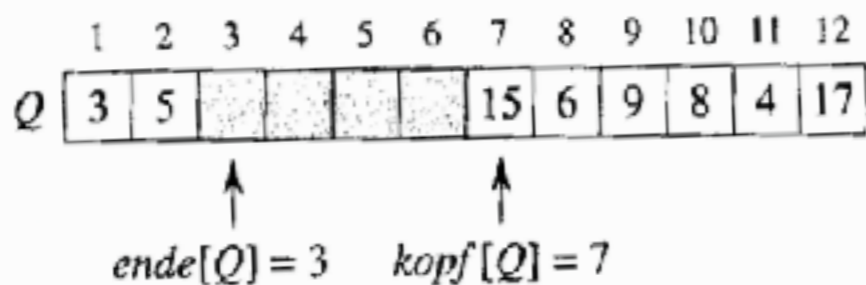
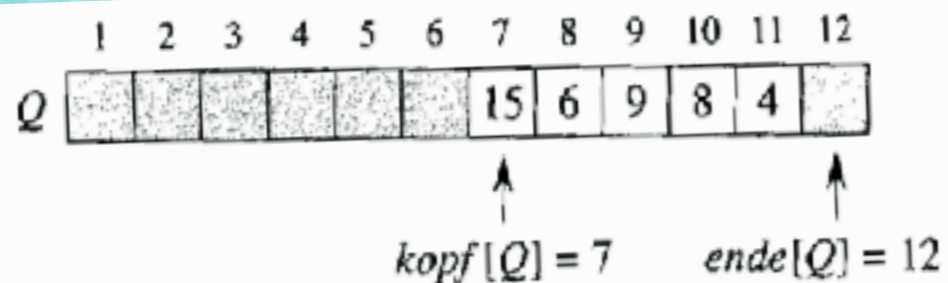
ENQUEUE: 17, 3, 5



DEQUEUE:



# WARTESCHLANGE AUF ARRAY UMGESETZT



ENQUEUE(Q, x)

```

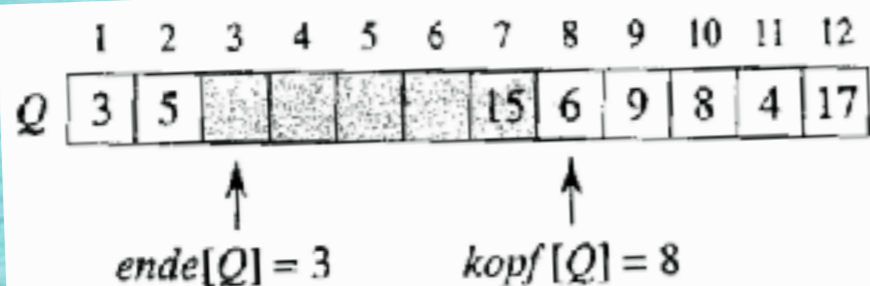
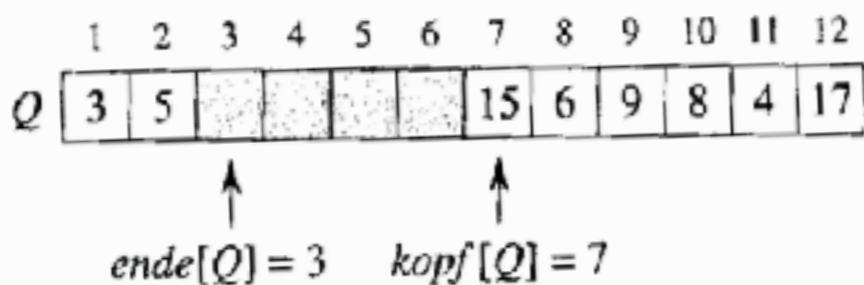
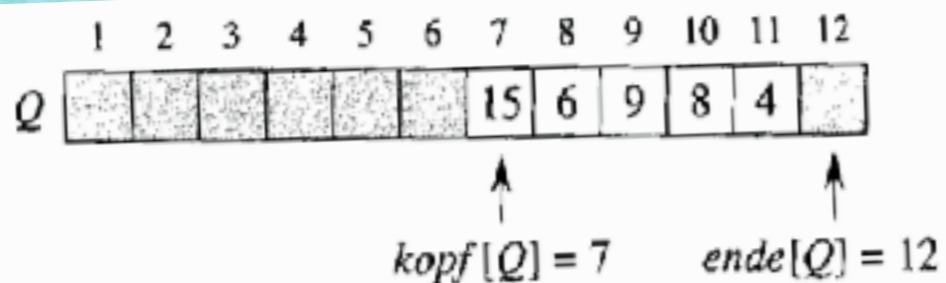
1 Q[ende[Q]] ← x
2 if ende[Q] = länge[Q]
3   then ende[Q] ← 1
4   else ende[Q] ← ende[Q] + 1
  
```

DEQUEUE(Q)

```

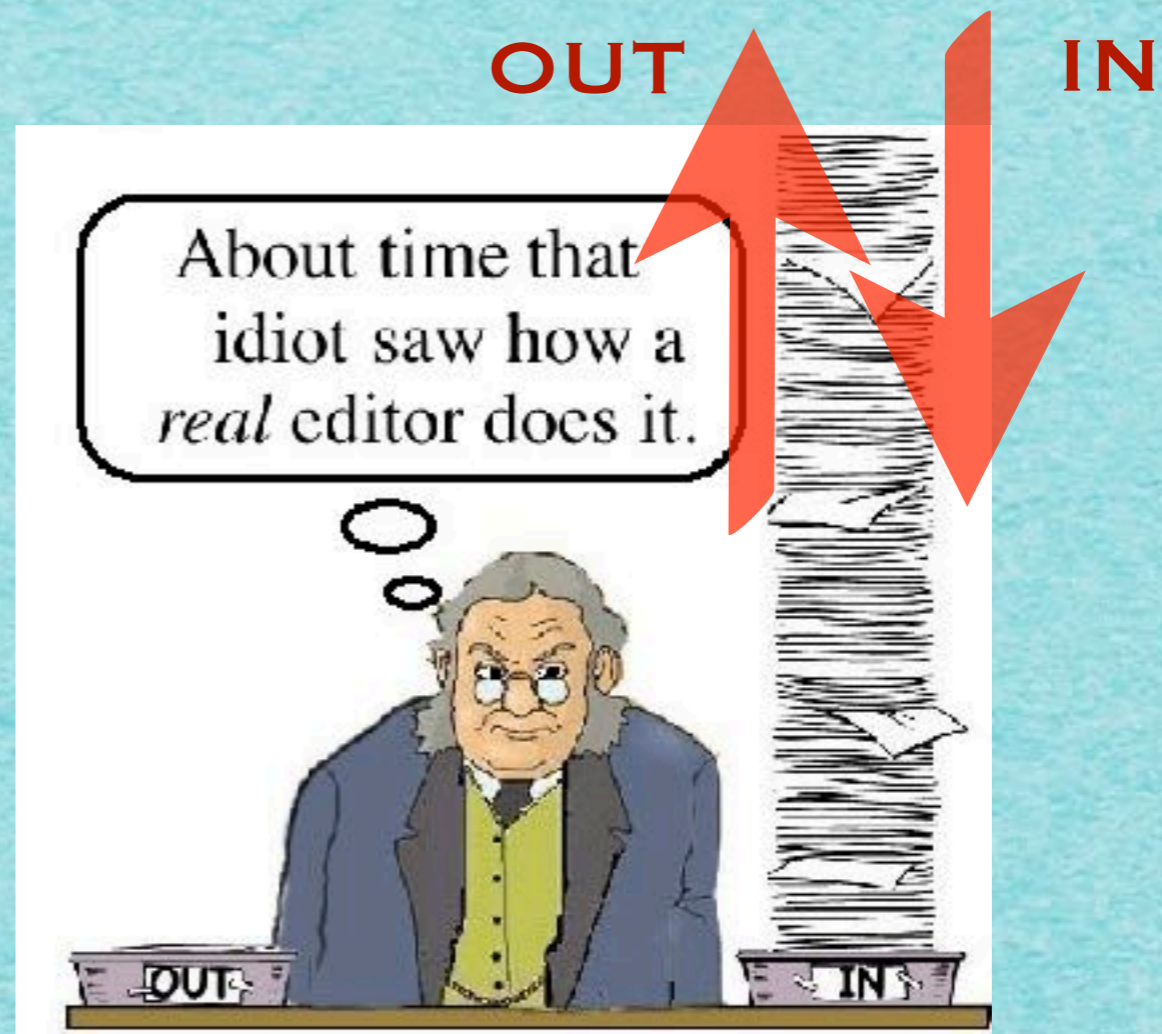
1 x ← Q[kopf[Q]]
2 if kopf[Q] = länge[Q]
3   then kopf[Q] ← 1
4   else kopf[Q] ← kopf[Q] + 1
5 return x
  
```

# WARTESCHLANGE AUF ARRAY UMGESETZT



# DATENSTRUKTUR II

R



**STAPEL: LAST IN - FIRST OUT**

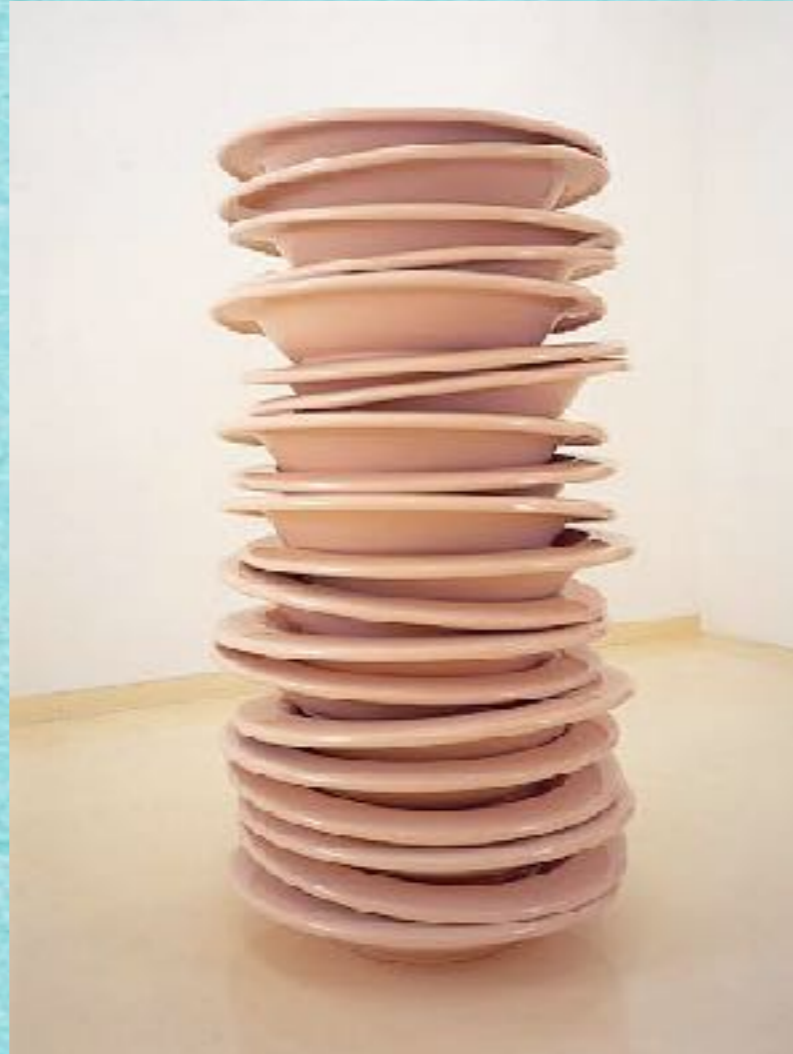


# DATENSTRUKTUR II



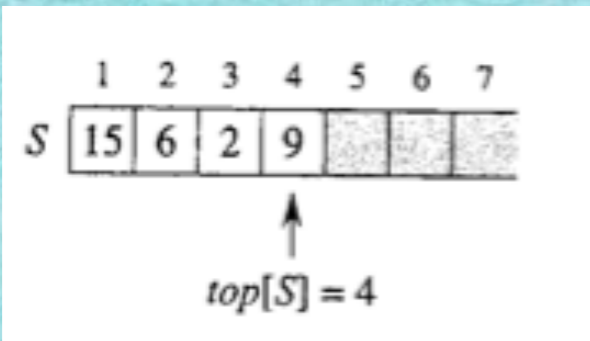
**STAPEL: LAST IN - FIRST OUT**

# DATENSTRUKTUR II

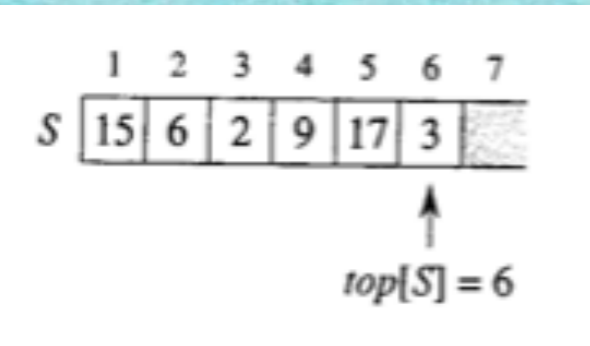


**STAPEL: LAST IN - FIRST OUT**

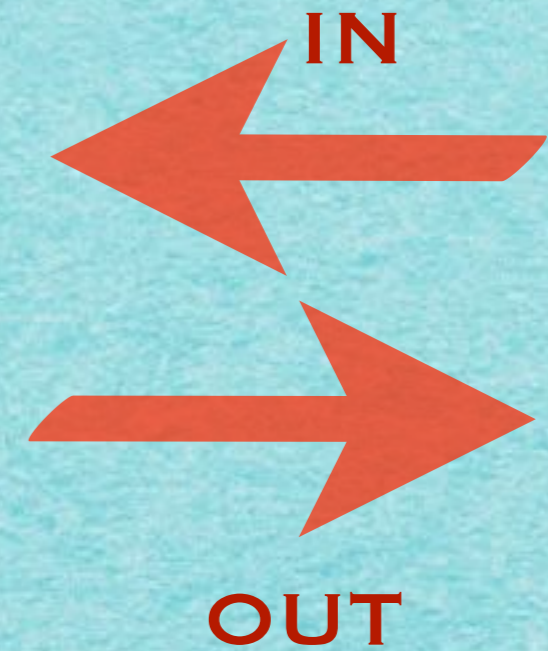
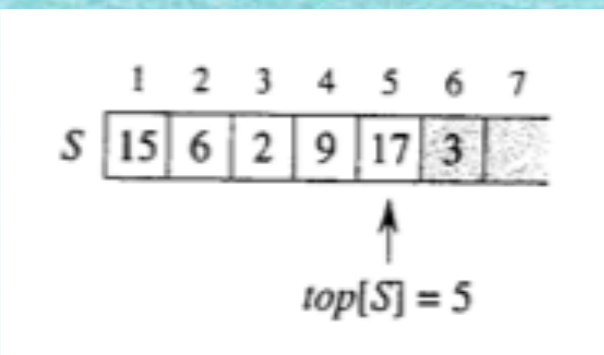
# STACK AUF ARRAY UMGESSETZT



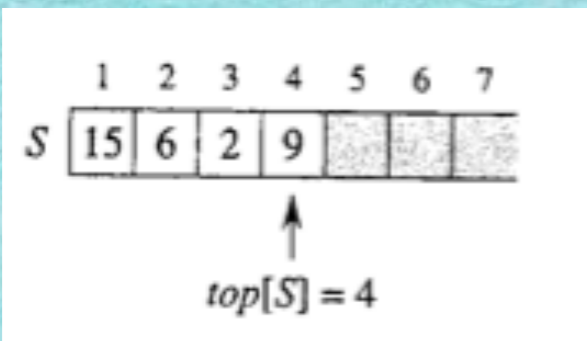
PUSH: 17, 3



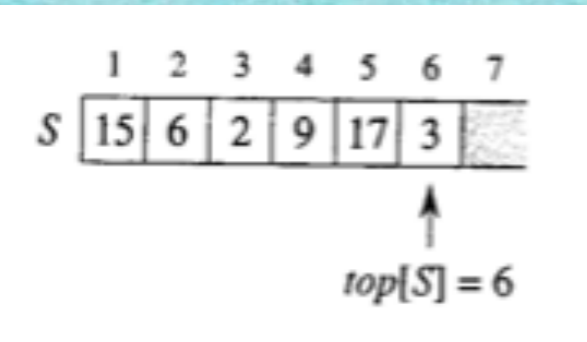
POP



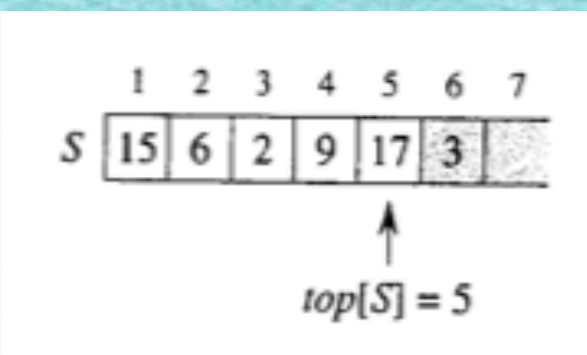
# STACK AUF ARRAY UMGESETZT



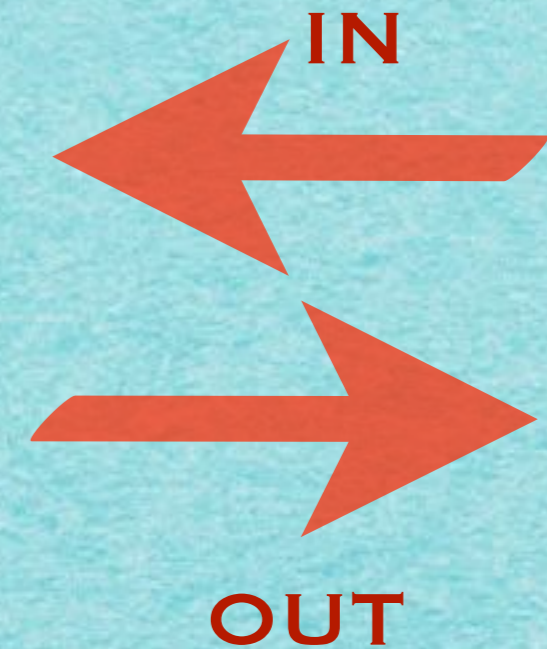
```
STACK-EMPTY( $S$ )  
1  if  $top[S] = 0$   
2    then return WAHR  
3    else return FALSCH
```



```
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 | Markieren

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!





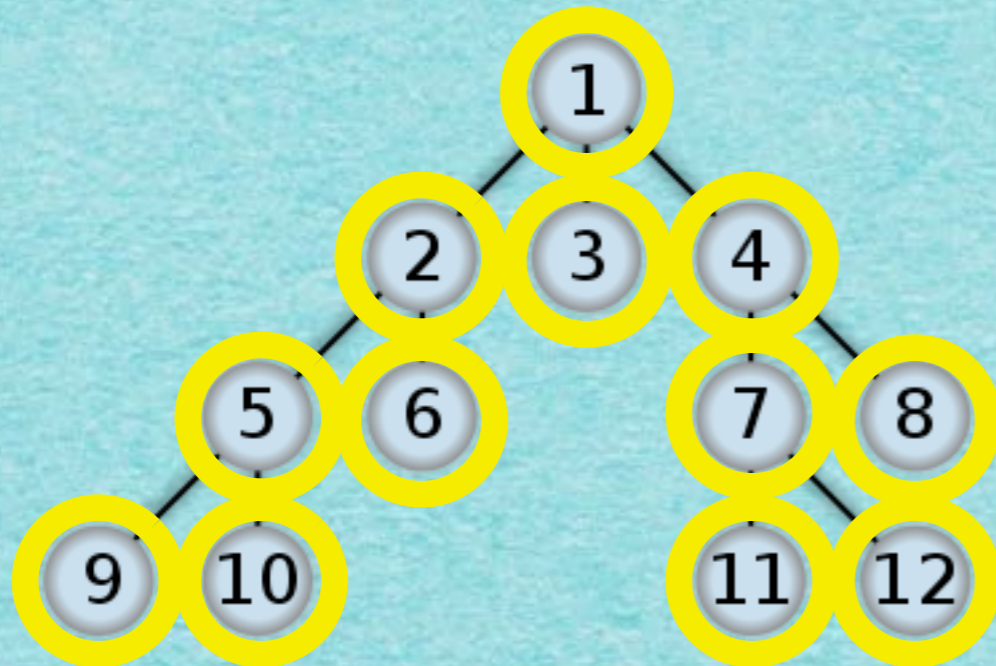
# *Kapitel 3.5: Tiefensuche und Breitensuche*

*Algorithmen und Datenstrukturen  
WS 2019/20*

**Prof. Dr. Sándor Fekete**



# Graphenscan mit WARTESCHLANGE

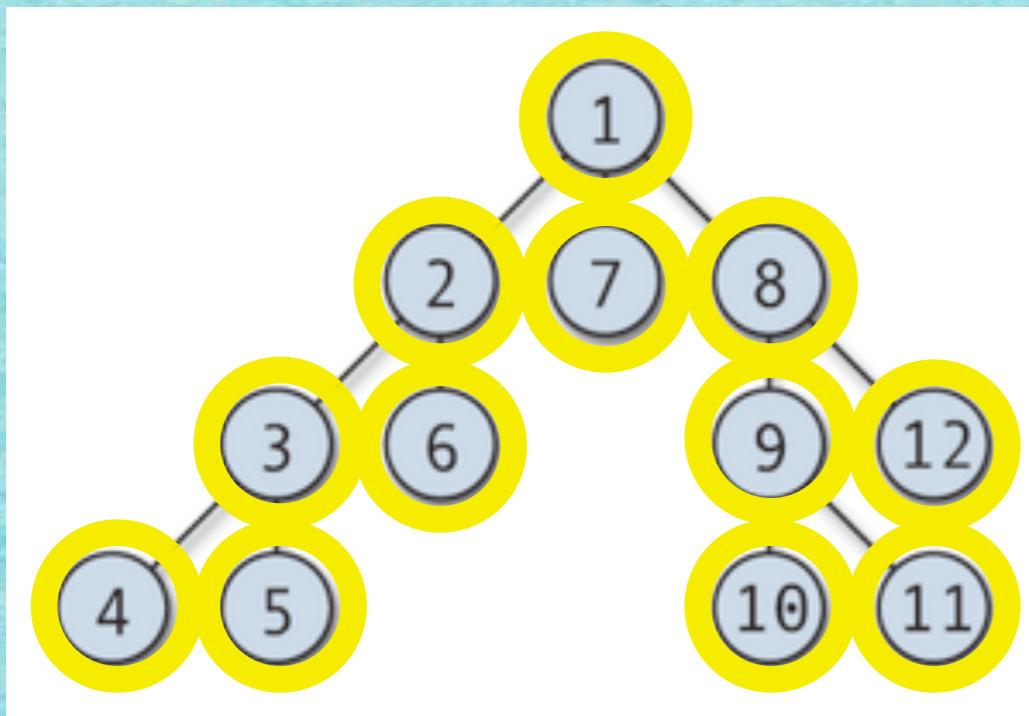


R:

**STOP!**

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

# Graphenscan mit STAPEL

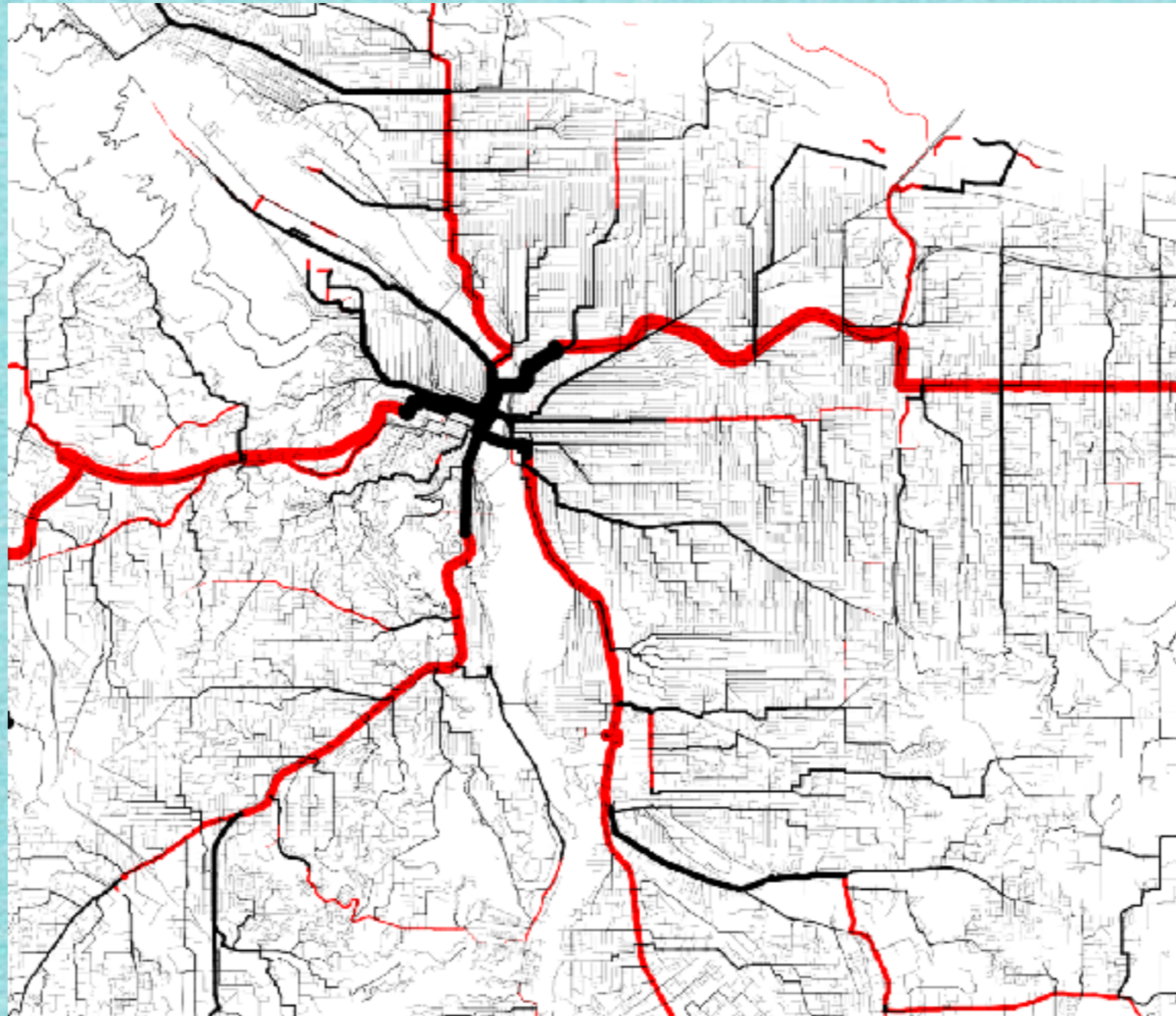


R:

**STOP!**

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

**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

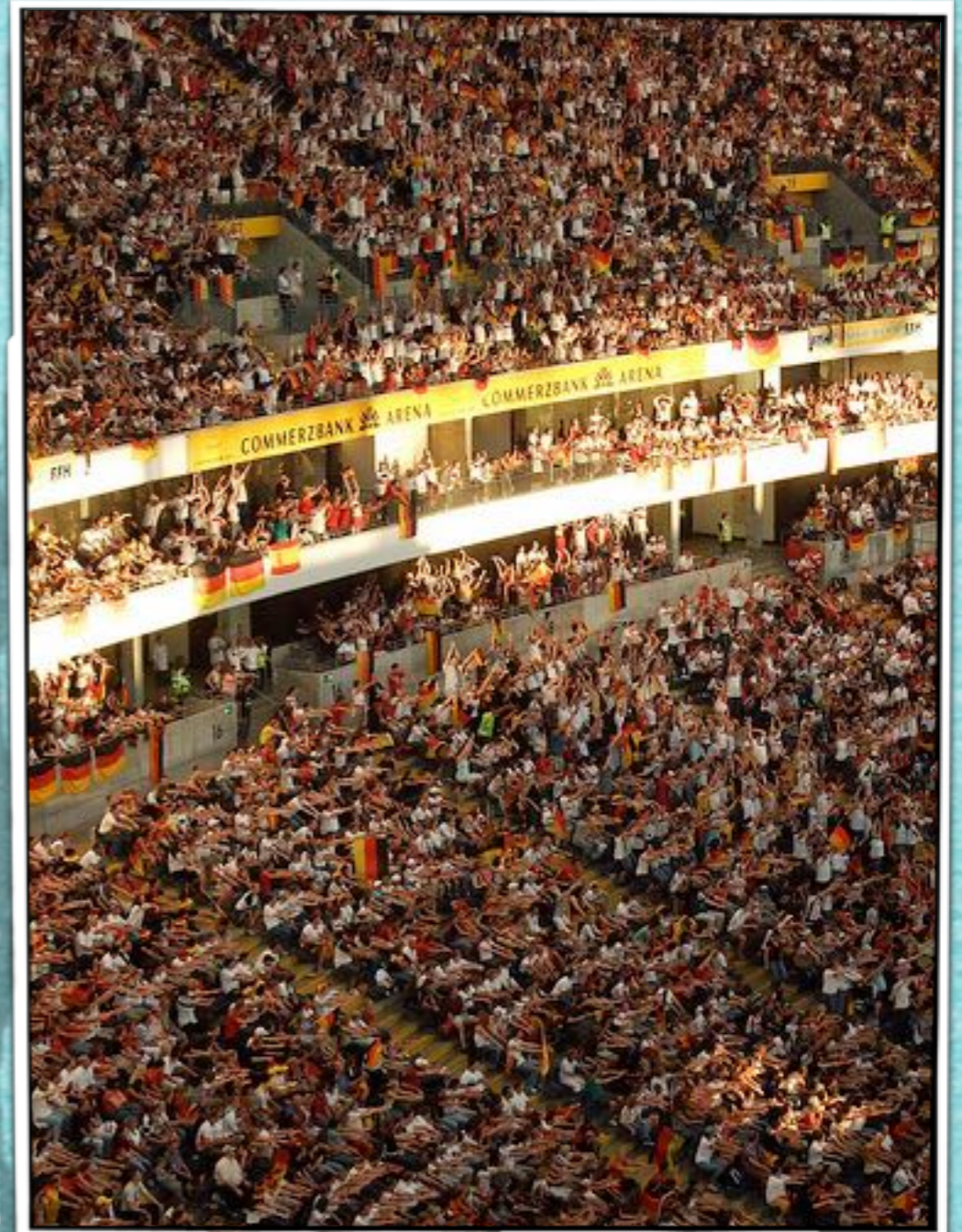
A. LOS bei „NULL“

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

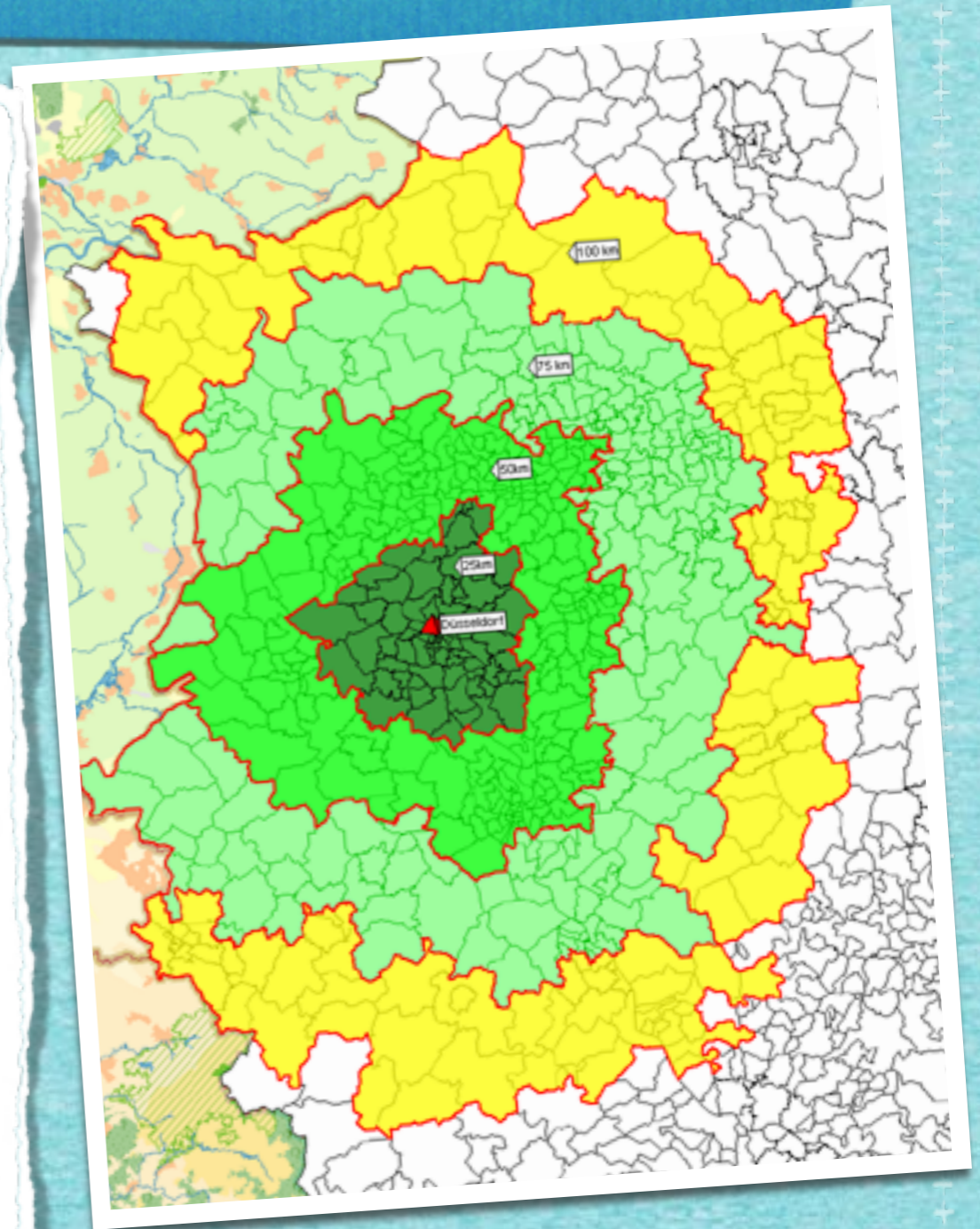
A. LOS bei „NULL“

B. Bis „ANGEKOMMEN!“:

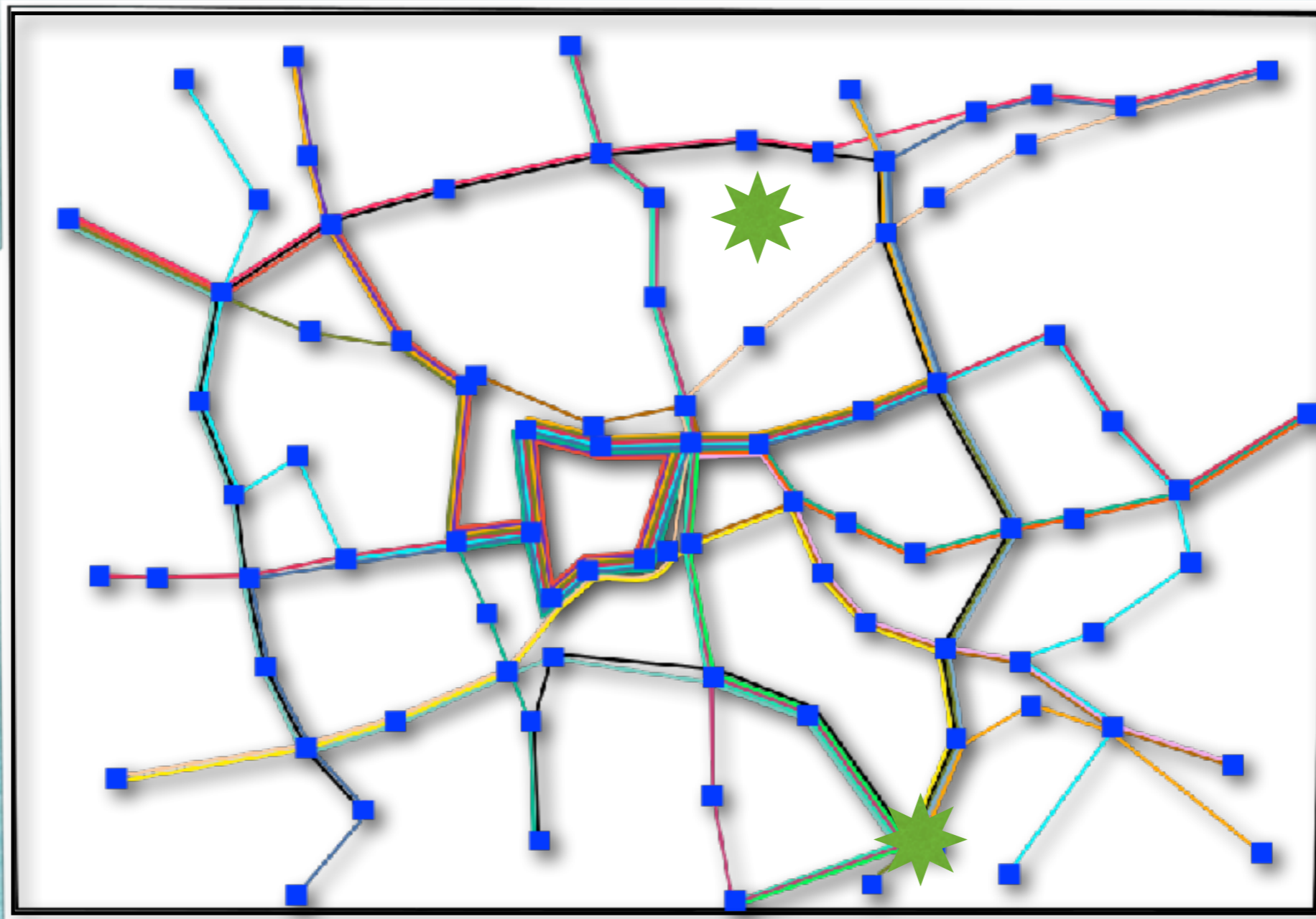
- Solange du noch nicht aufgestanden warst:
  - ▶ Wenn ein oder mehrere direkte Nachbarn aufstehen:
    1. Einen dieser Nachbarn merken
    2. In der nächsten Runde:
      - 2.1. aufstehen
      - 2.2. Zahl merken
    3. In der übernächsten Runde hinsetzen

C. Nach „ANGEKOMMEN!“:

- Auf gemerkten Nachbarn zeigen

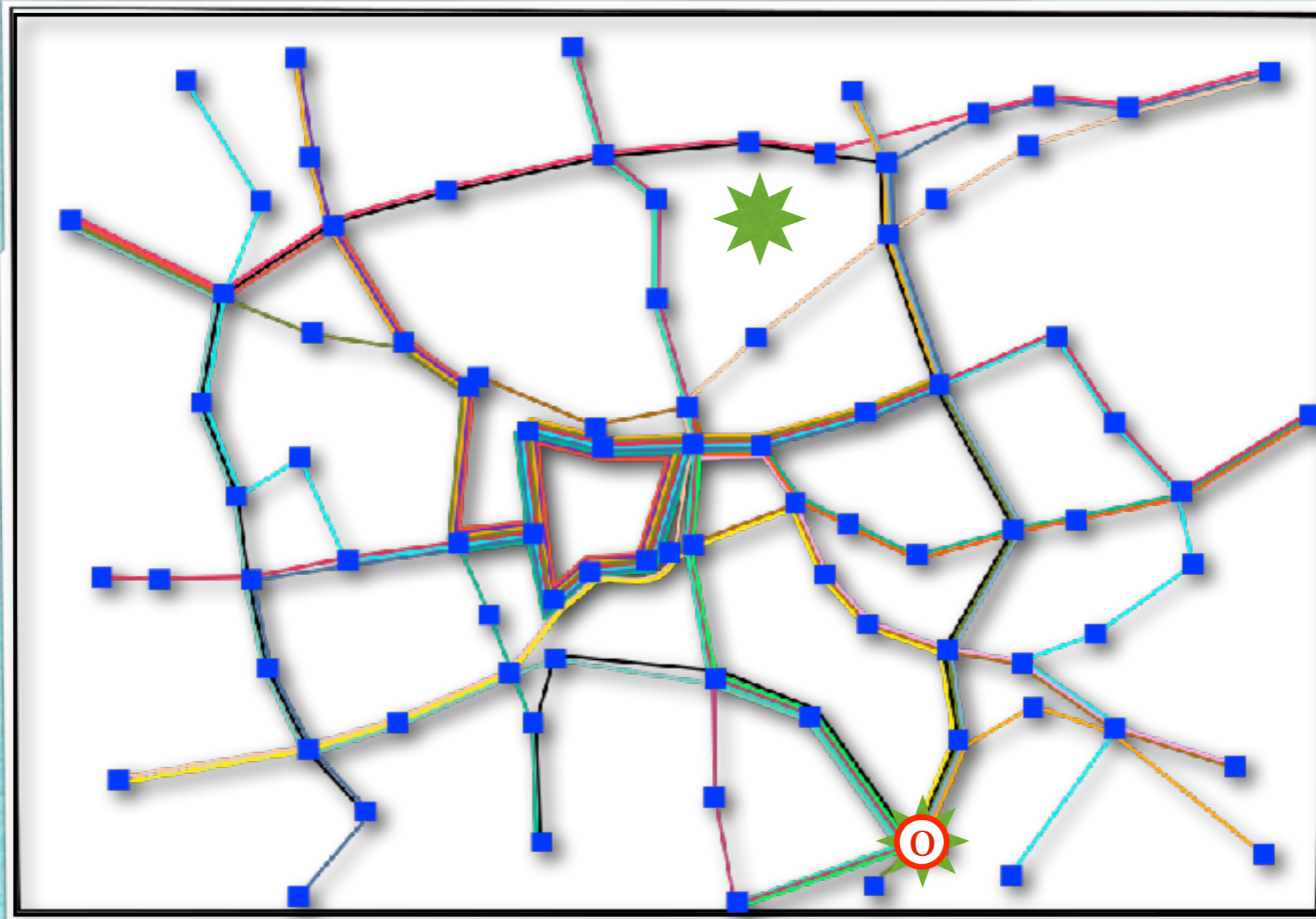


# Wellenreiten in Graphen

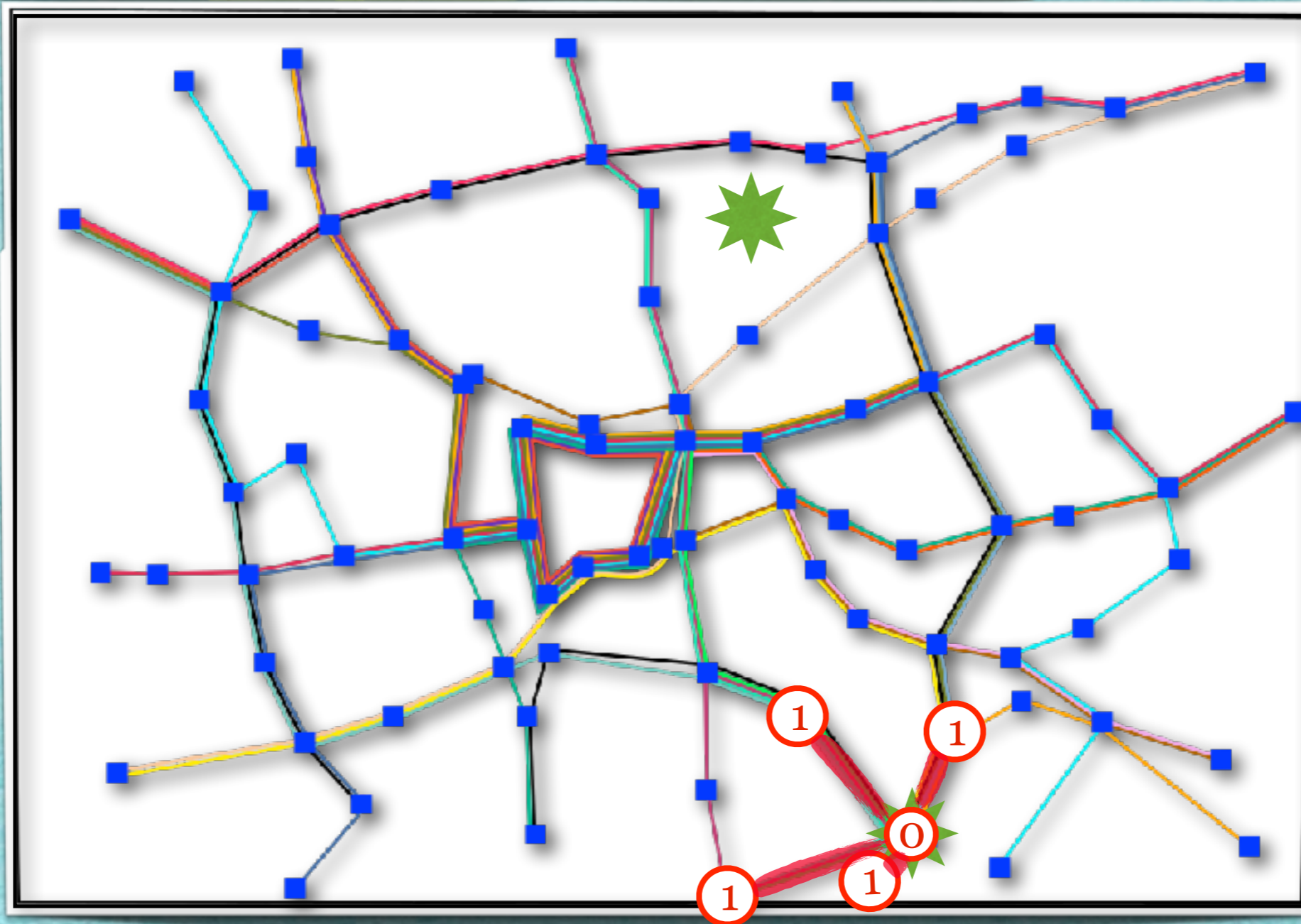




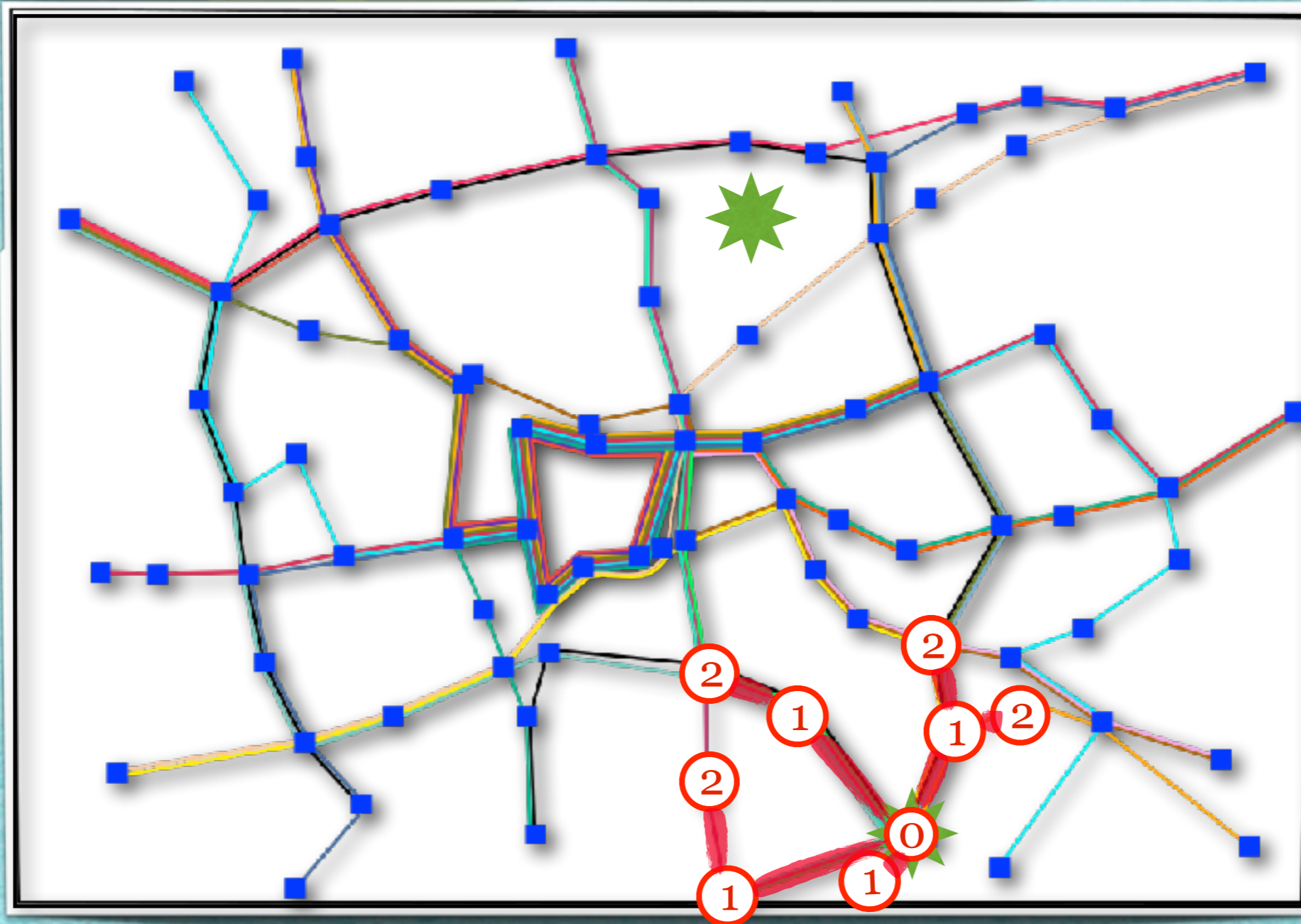
# Wellenreiten in Graphen



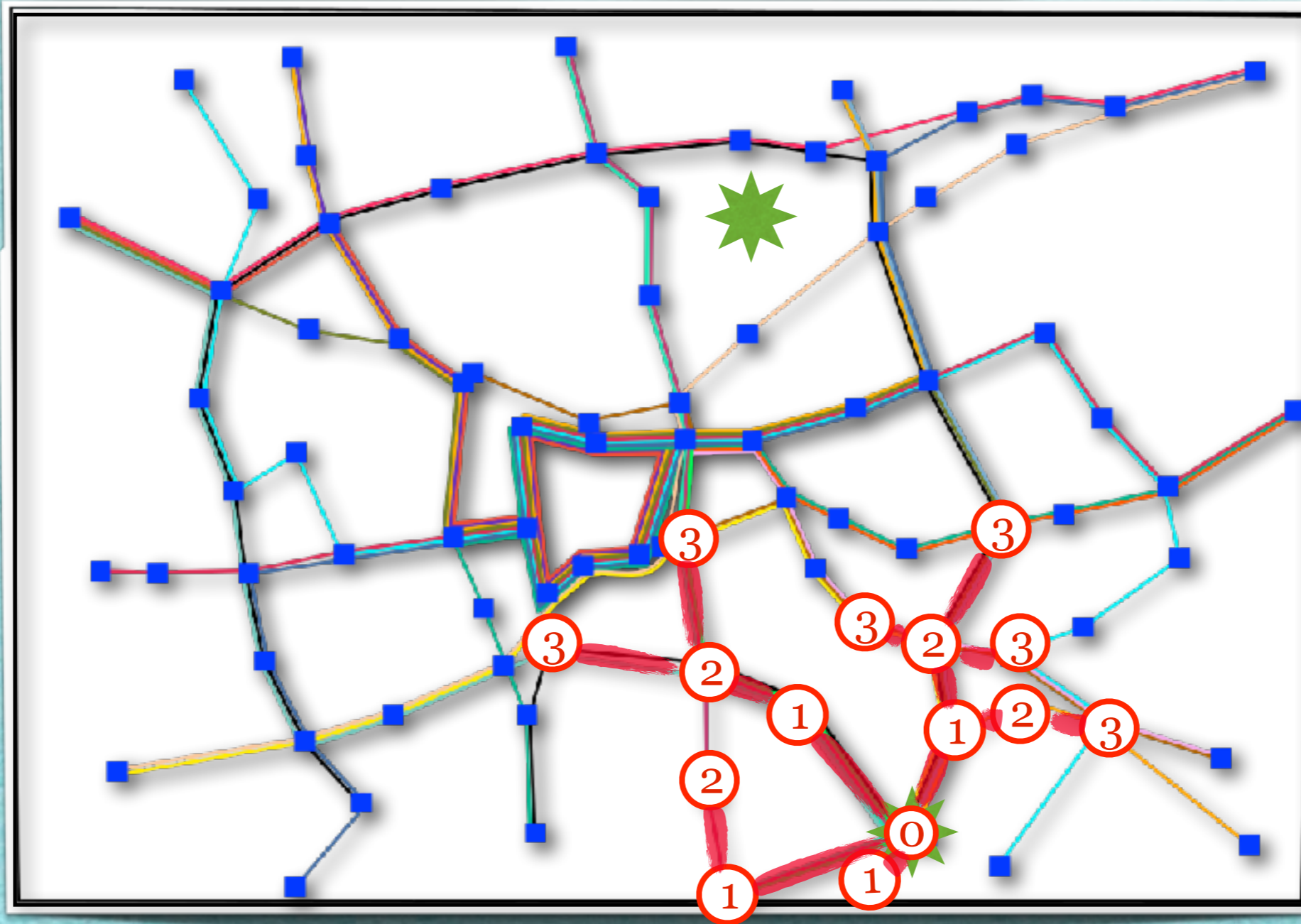
# Wellenreiten in Graphen



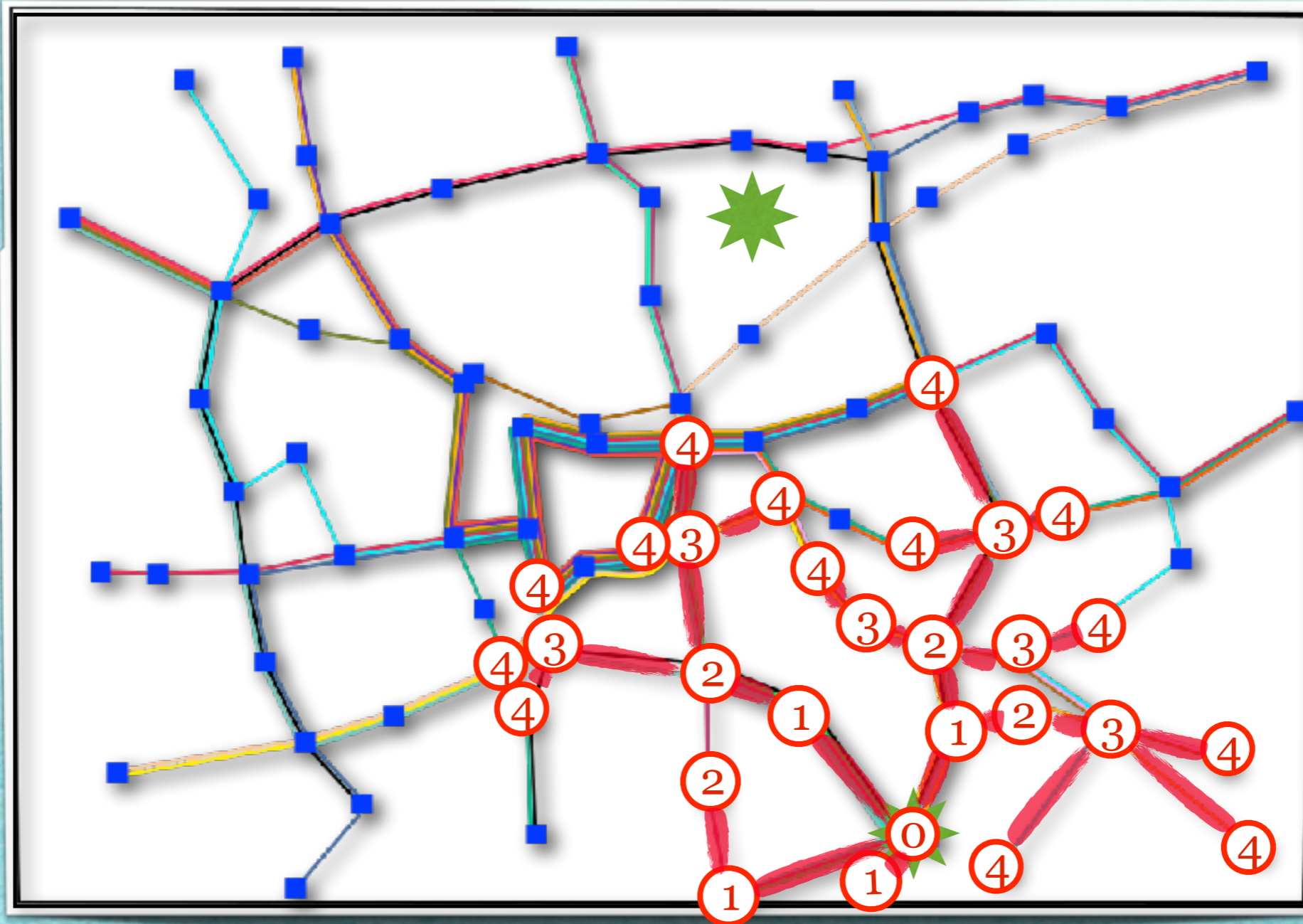
# Wellenreiten in Graphen



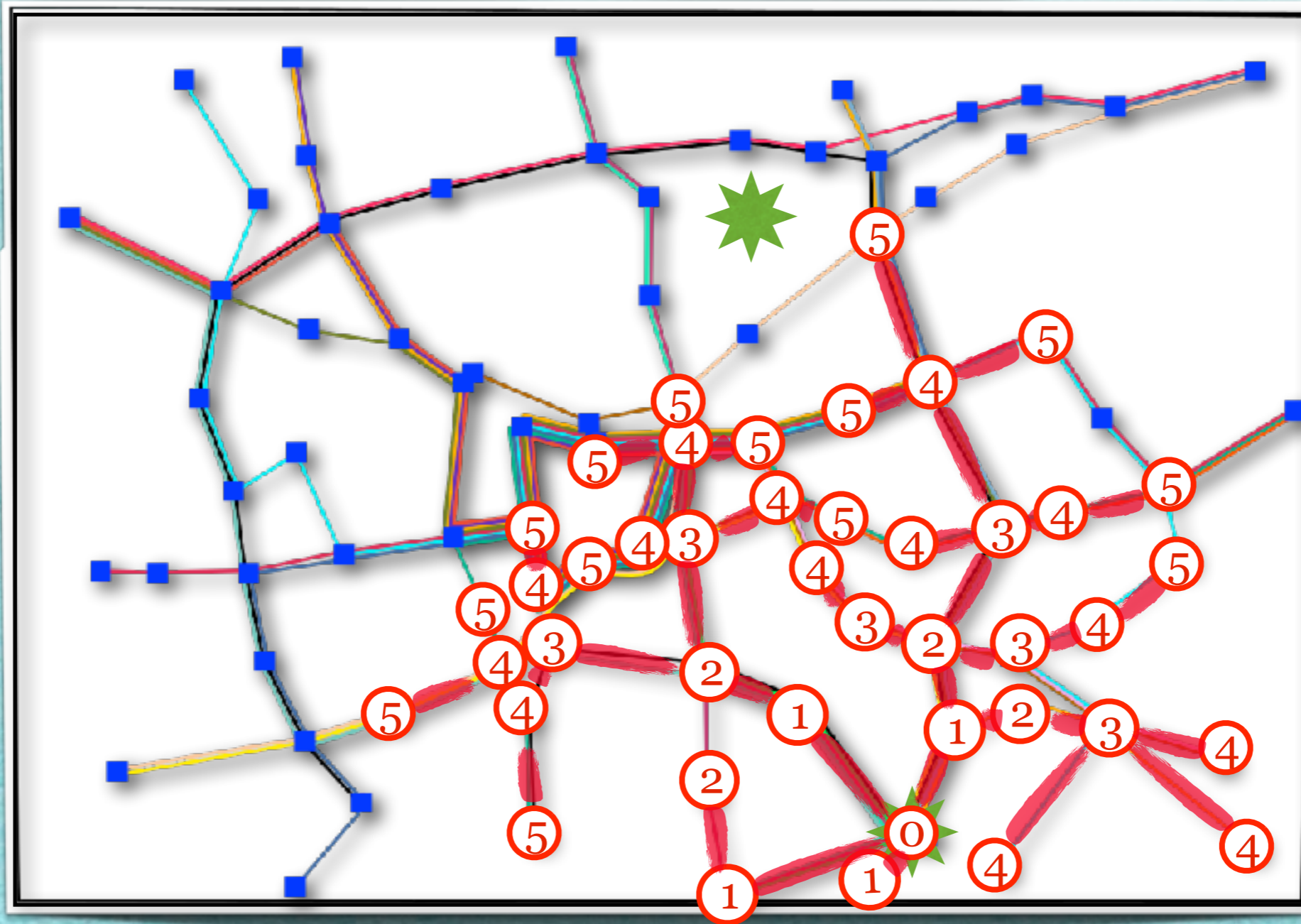
# Wellenreiten in Graphen



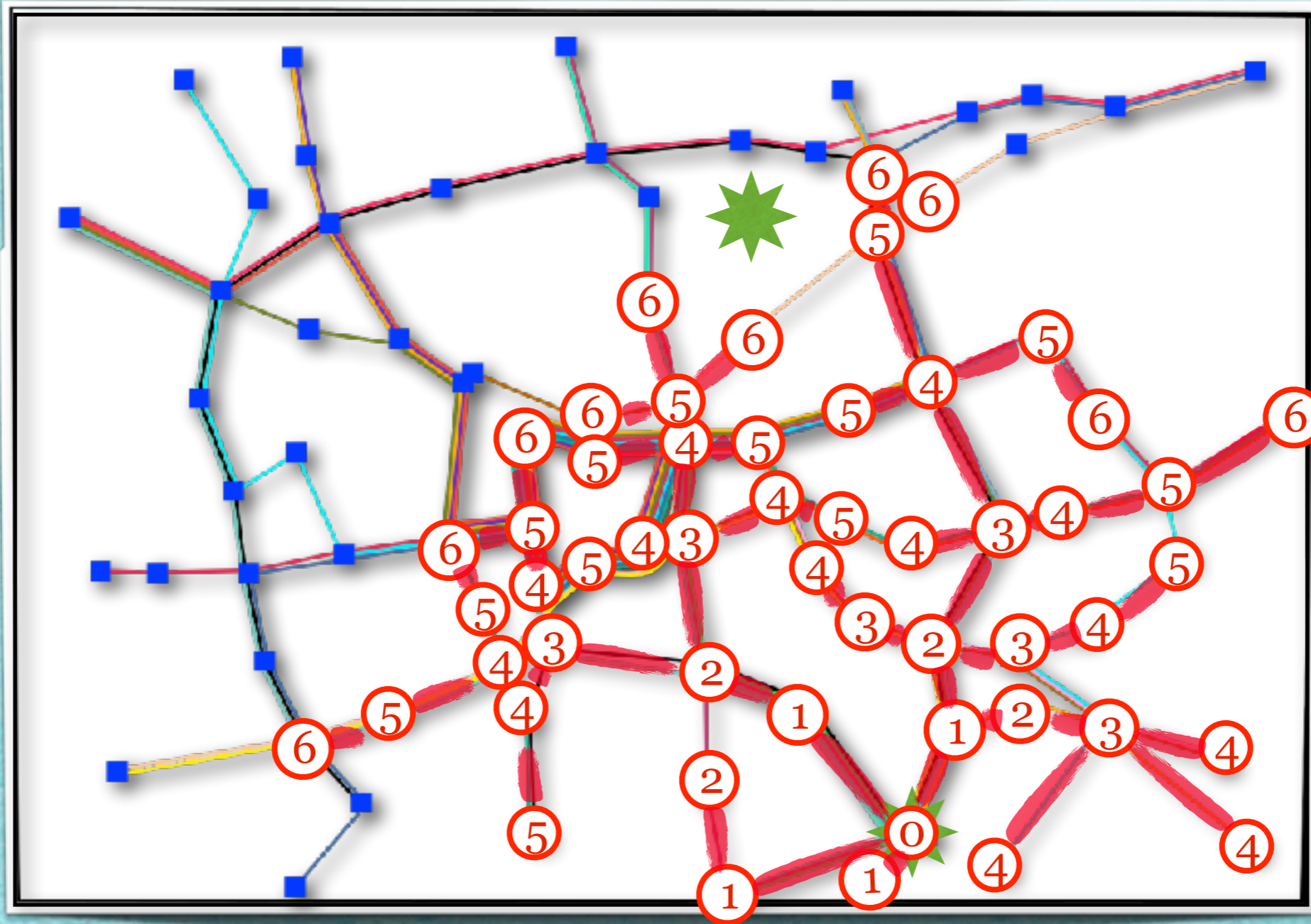
# Wellenreiten in Graphen



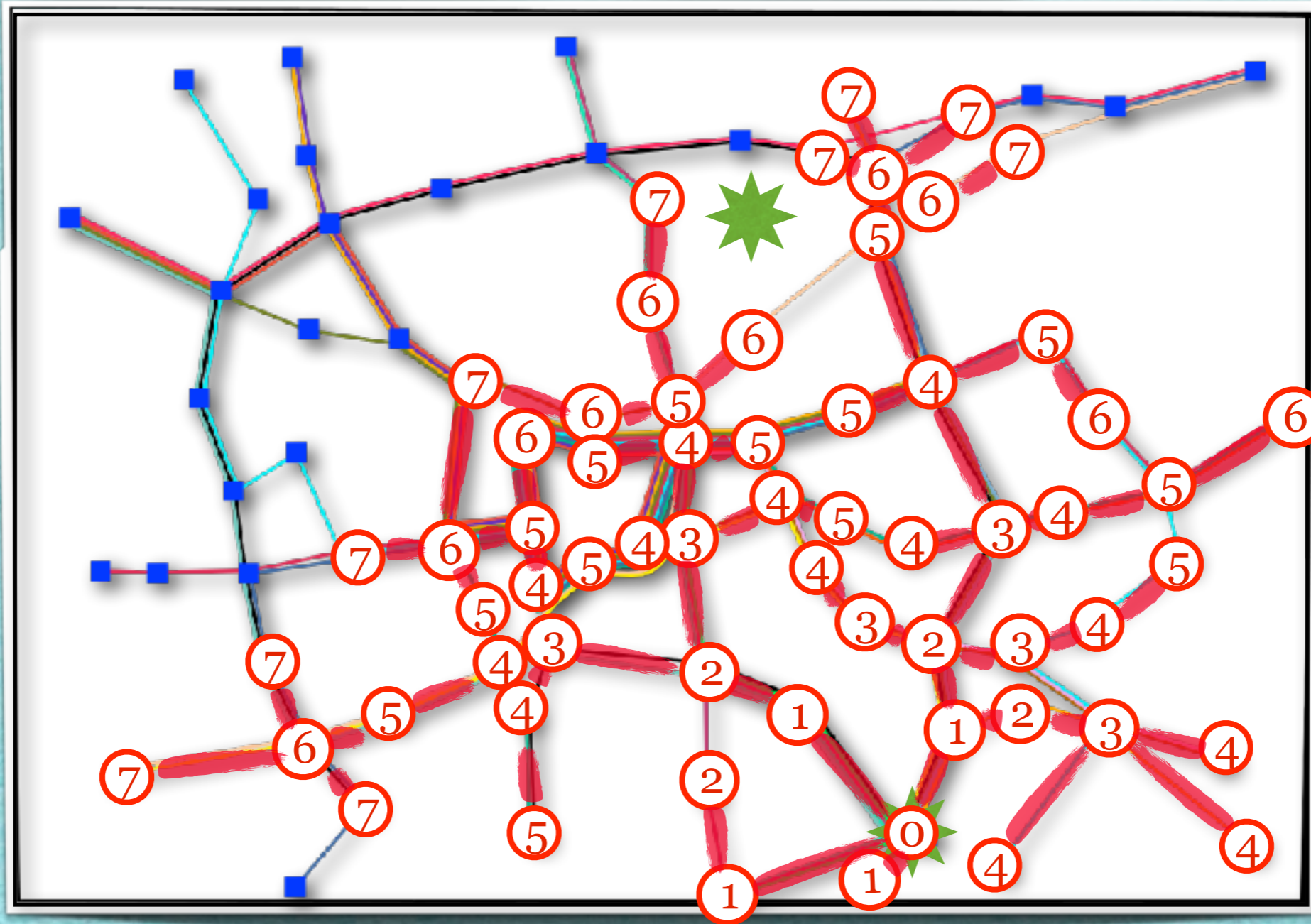
# Wellenreiten in Graphen



# Wellenreiten in Graphen

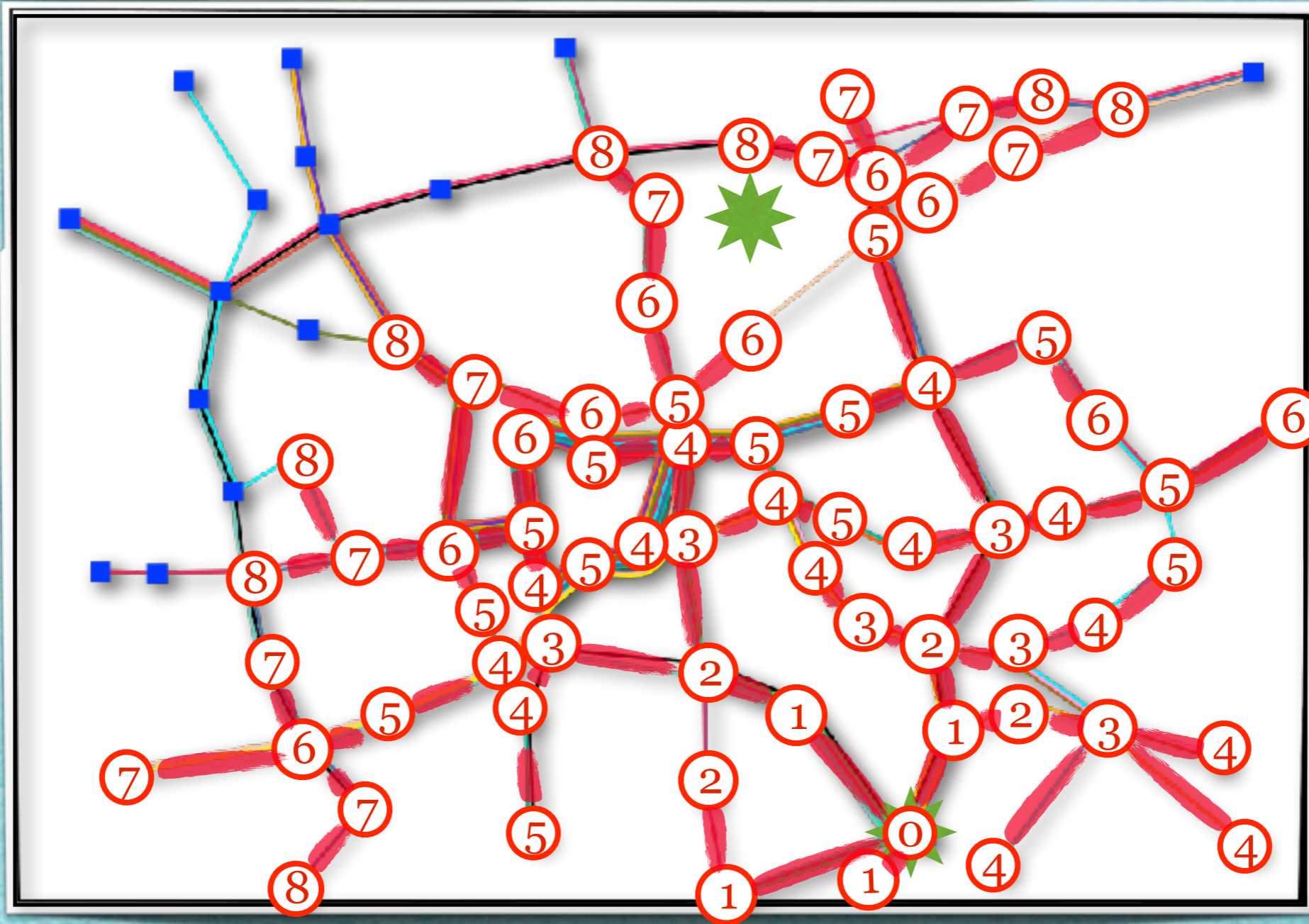


# Wellenreiten in Graphen

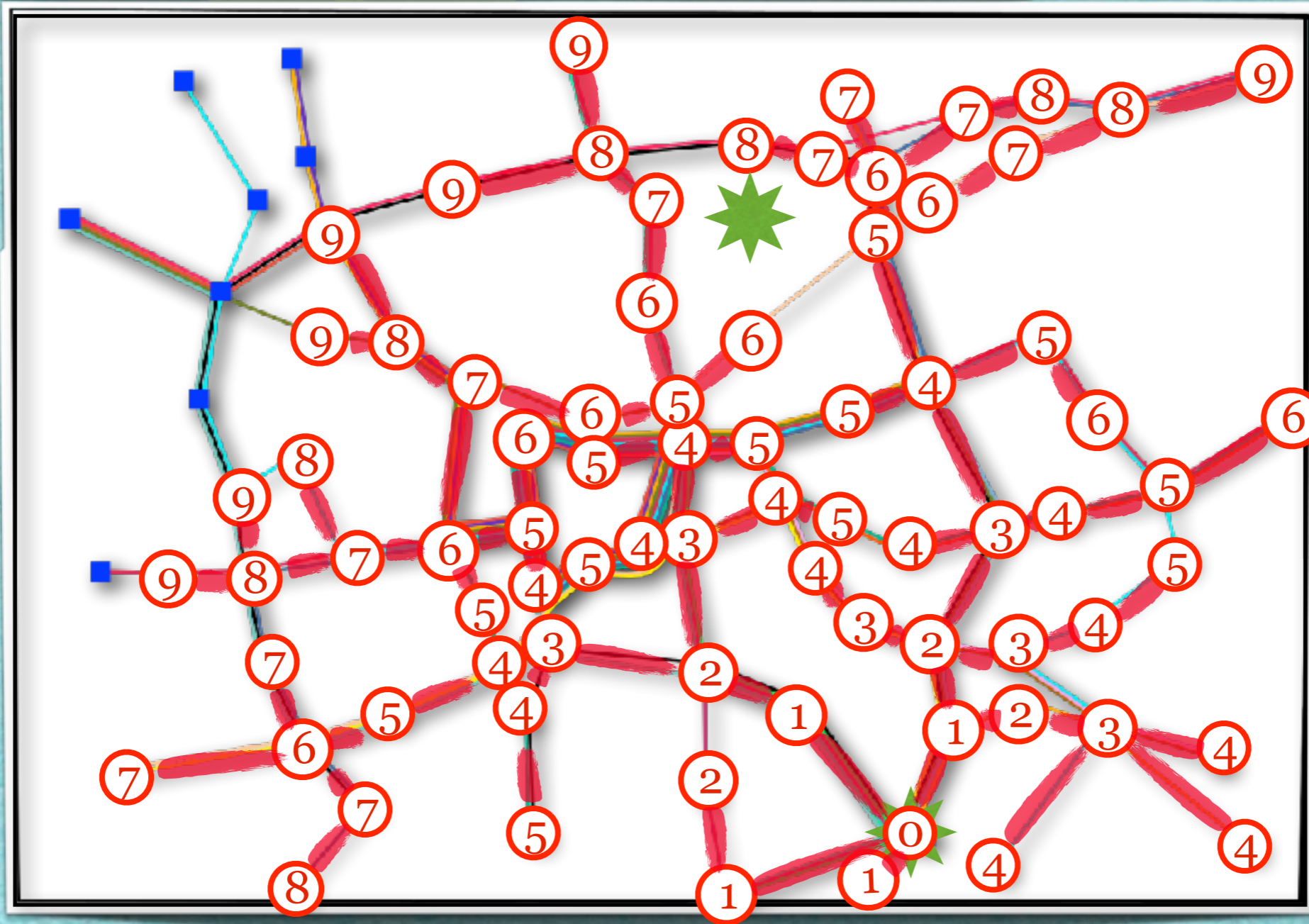




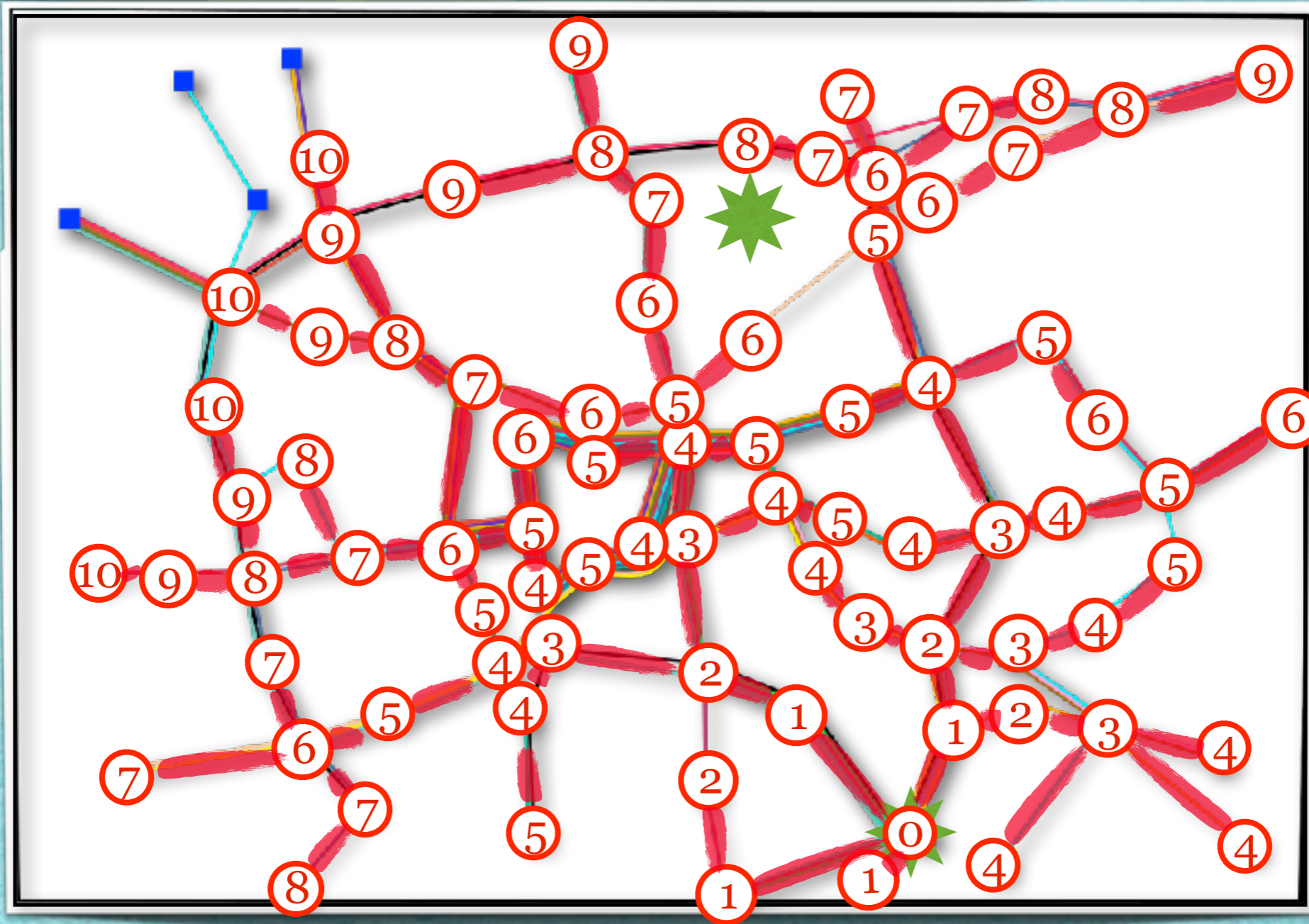
# Wellenreiten in Graphen



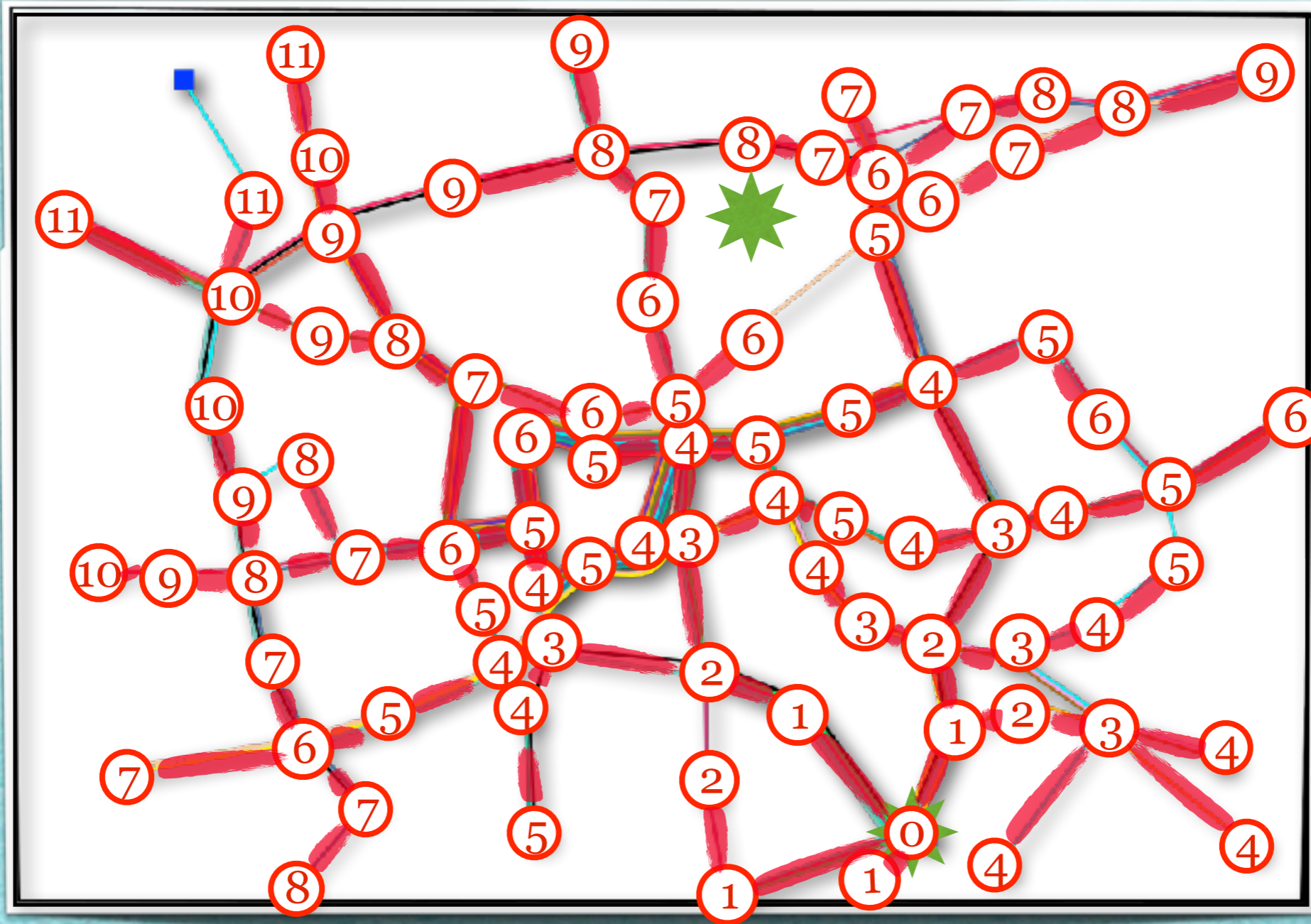
# Wellenreiten in Graphen



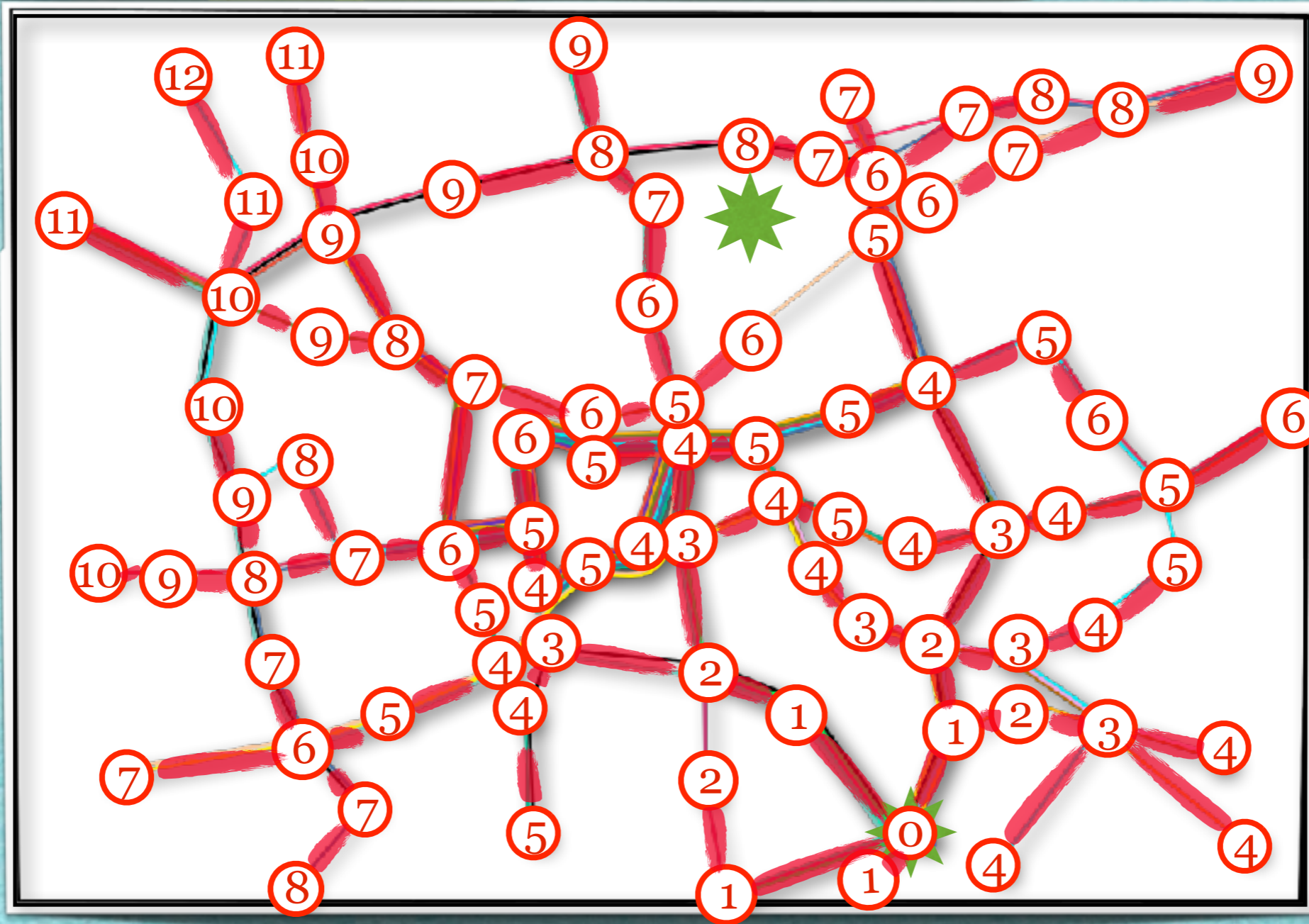
# Wellenreiten in Graphen



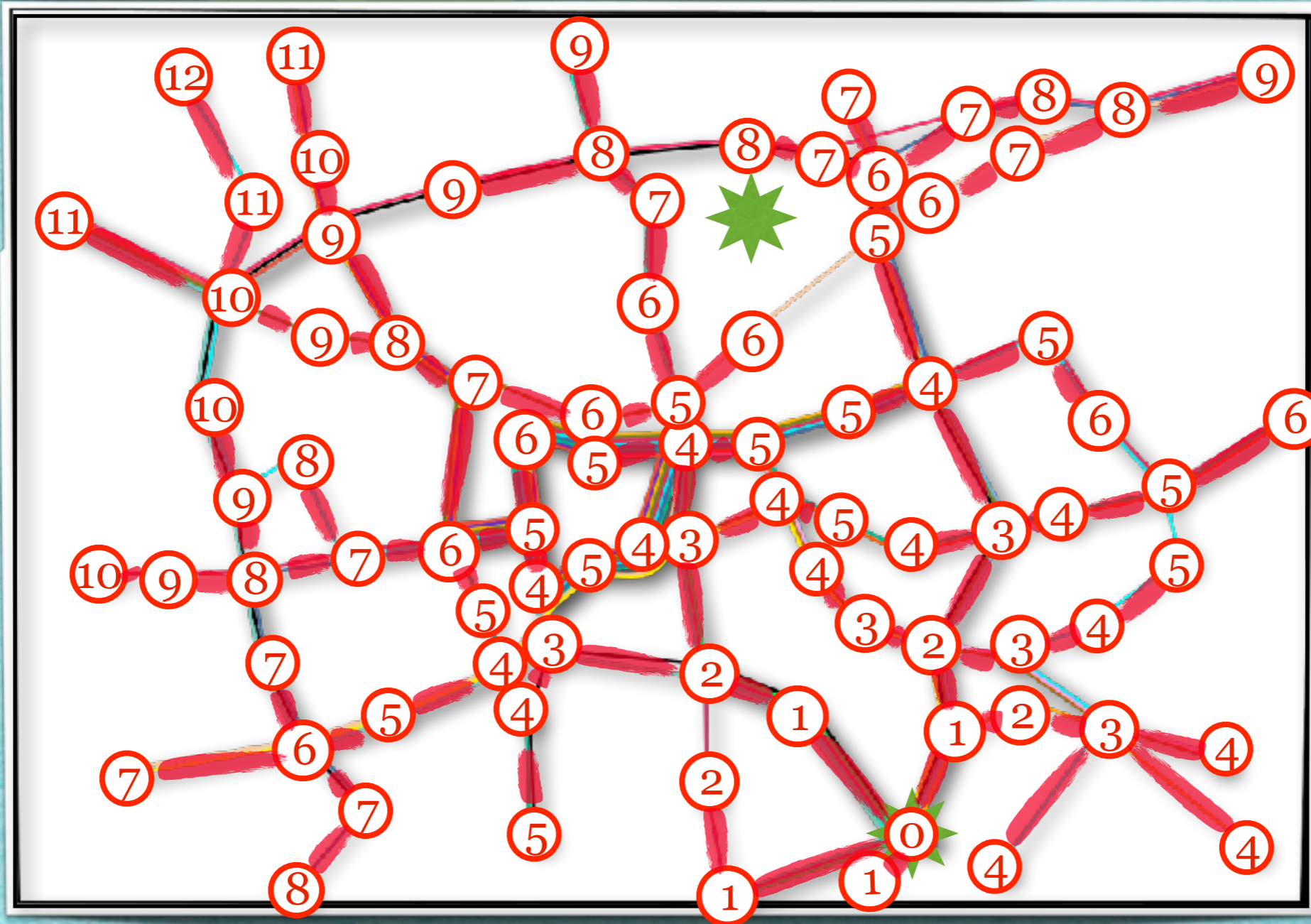
# Wellenreiten in Graphen



# Wellenreiten in Graphen

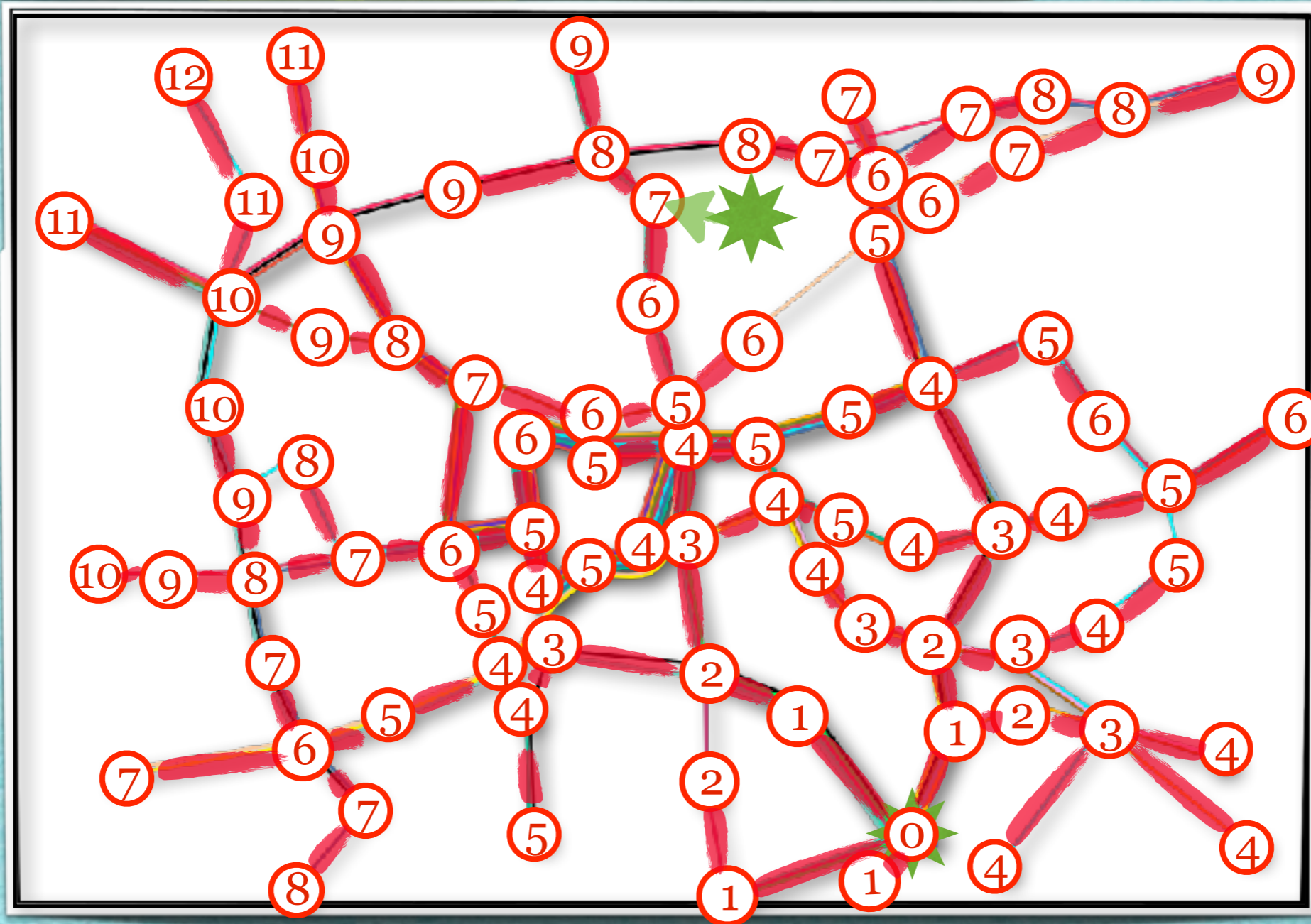


# Wellenreiten in Graphen



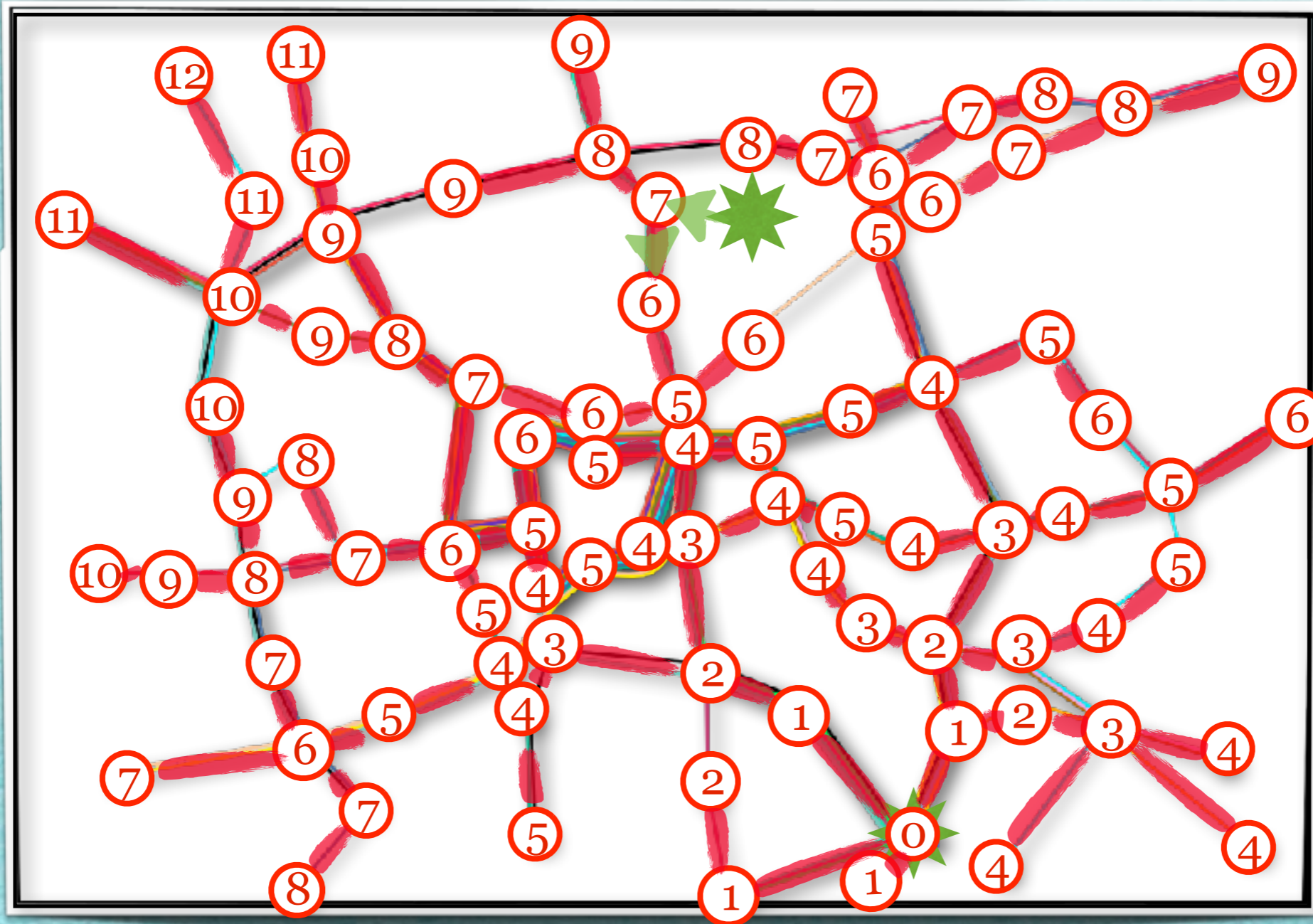
Breitensuche

# Wellenreiten in Graphen



Breitensuche

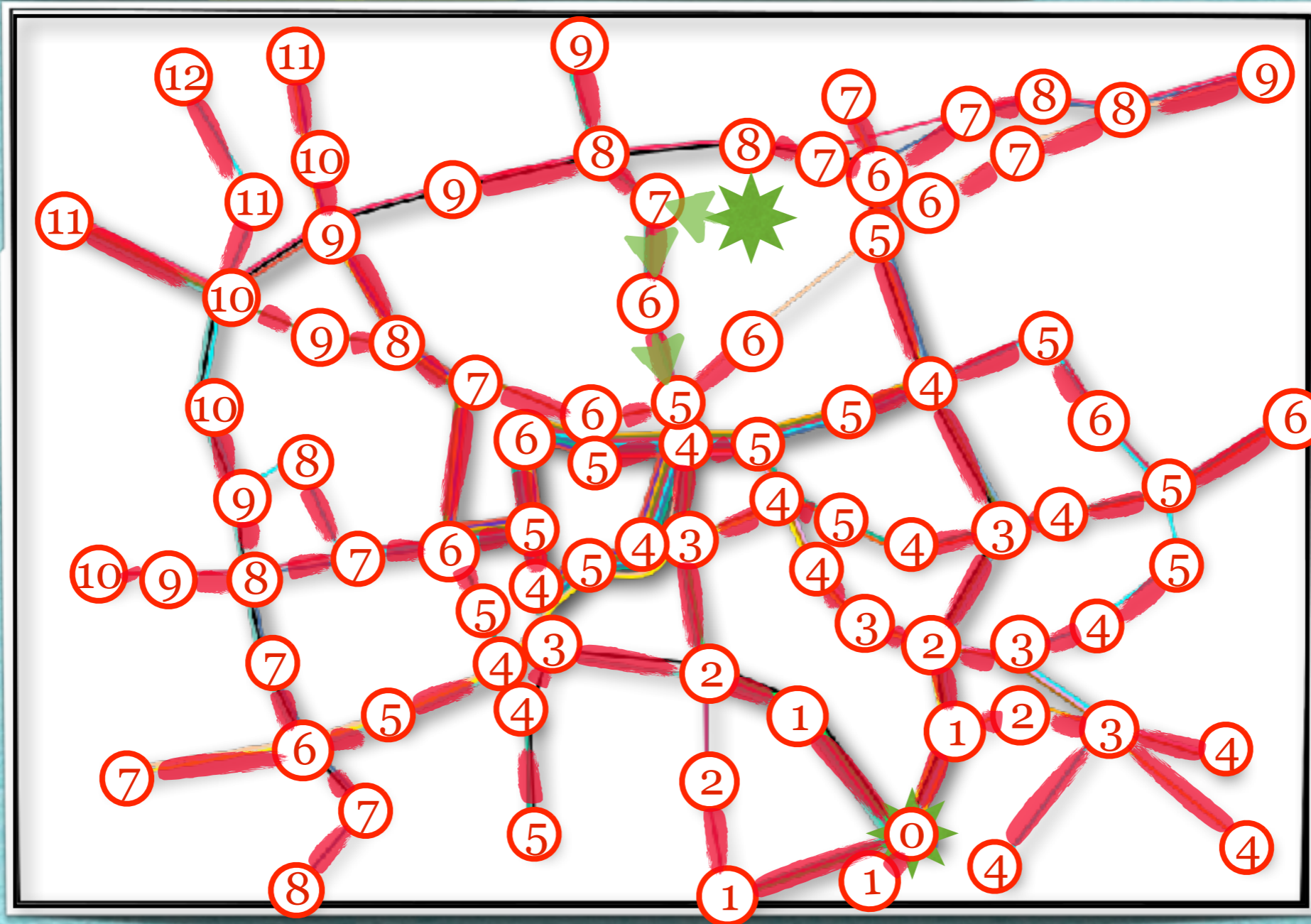
# Wellenreiten in Graphen



Breitensuche

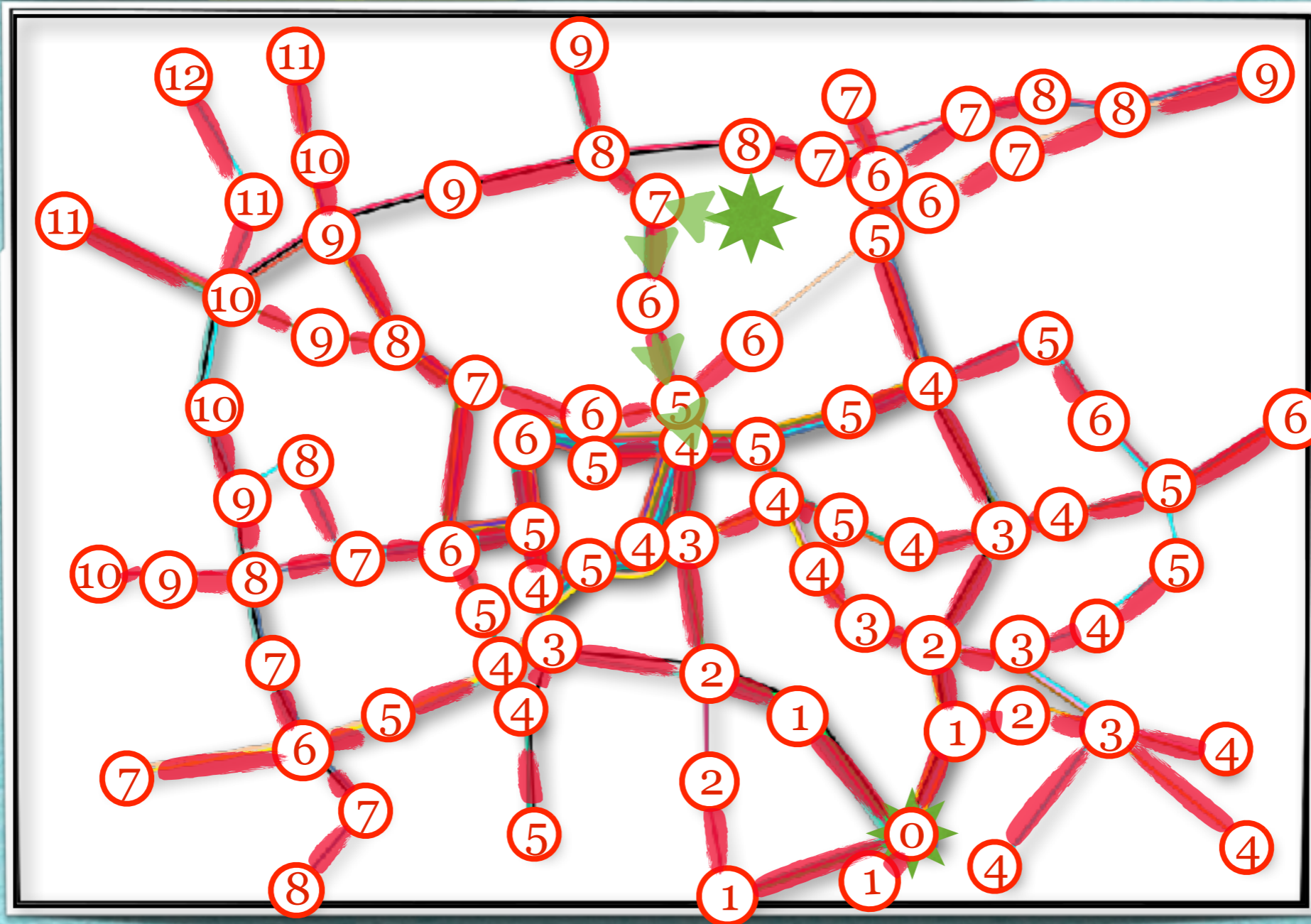


# Wellenreiten in Graphen



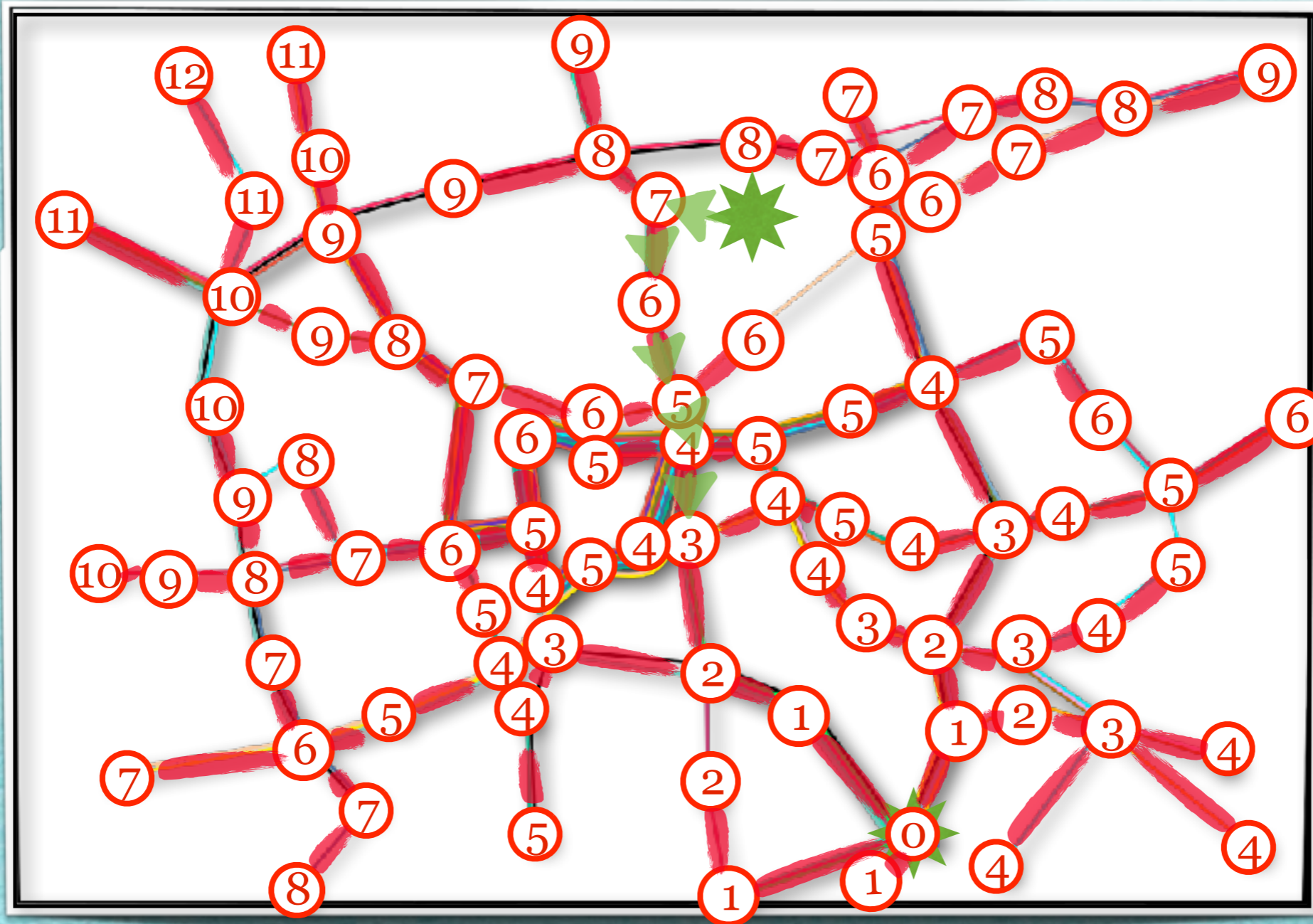
Breitensuche

# Wellenreiten in Graphen



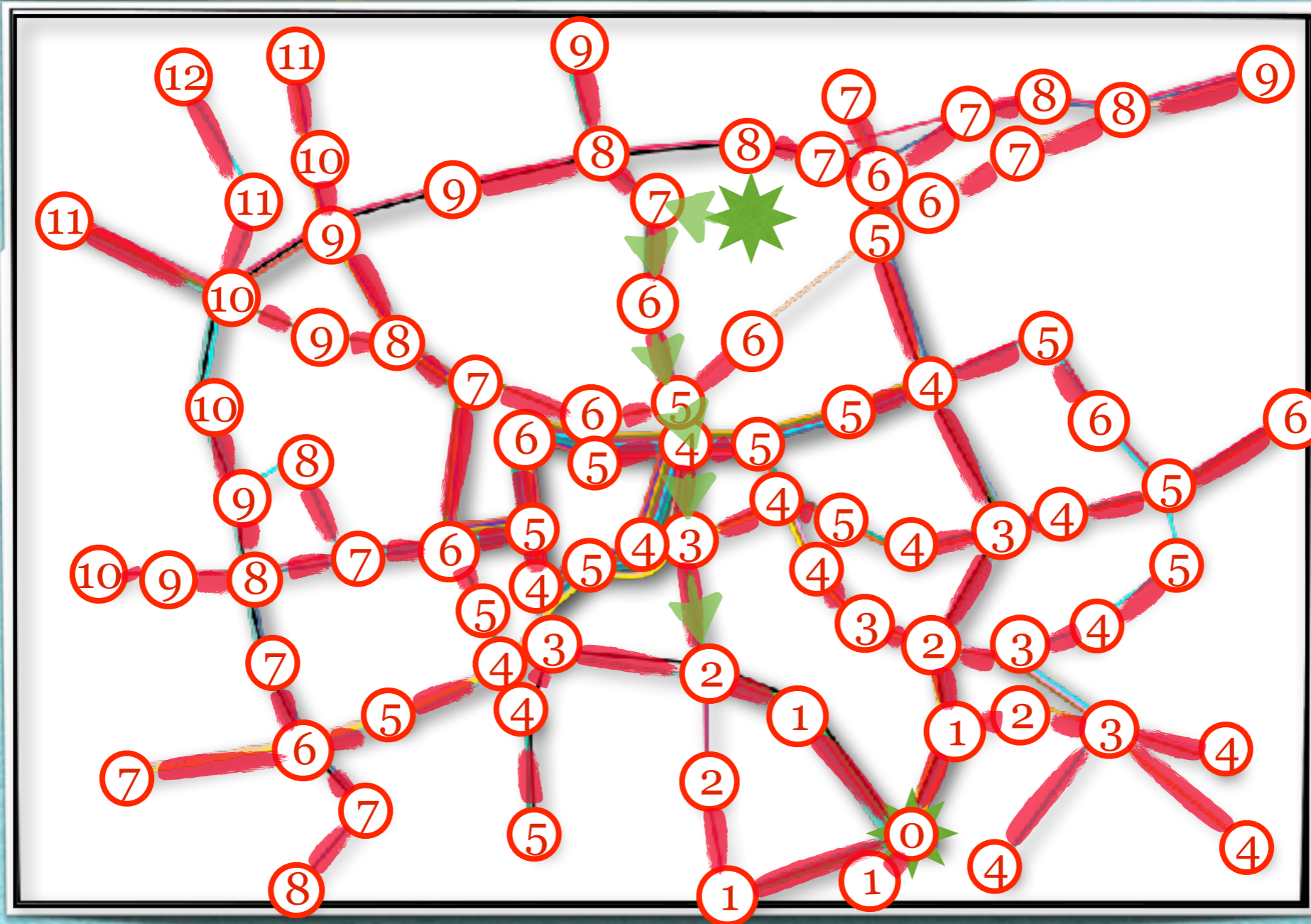
Breitensuche

# Wellenreiten in Graphen



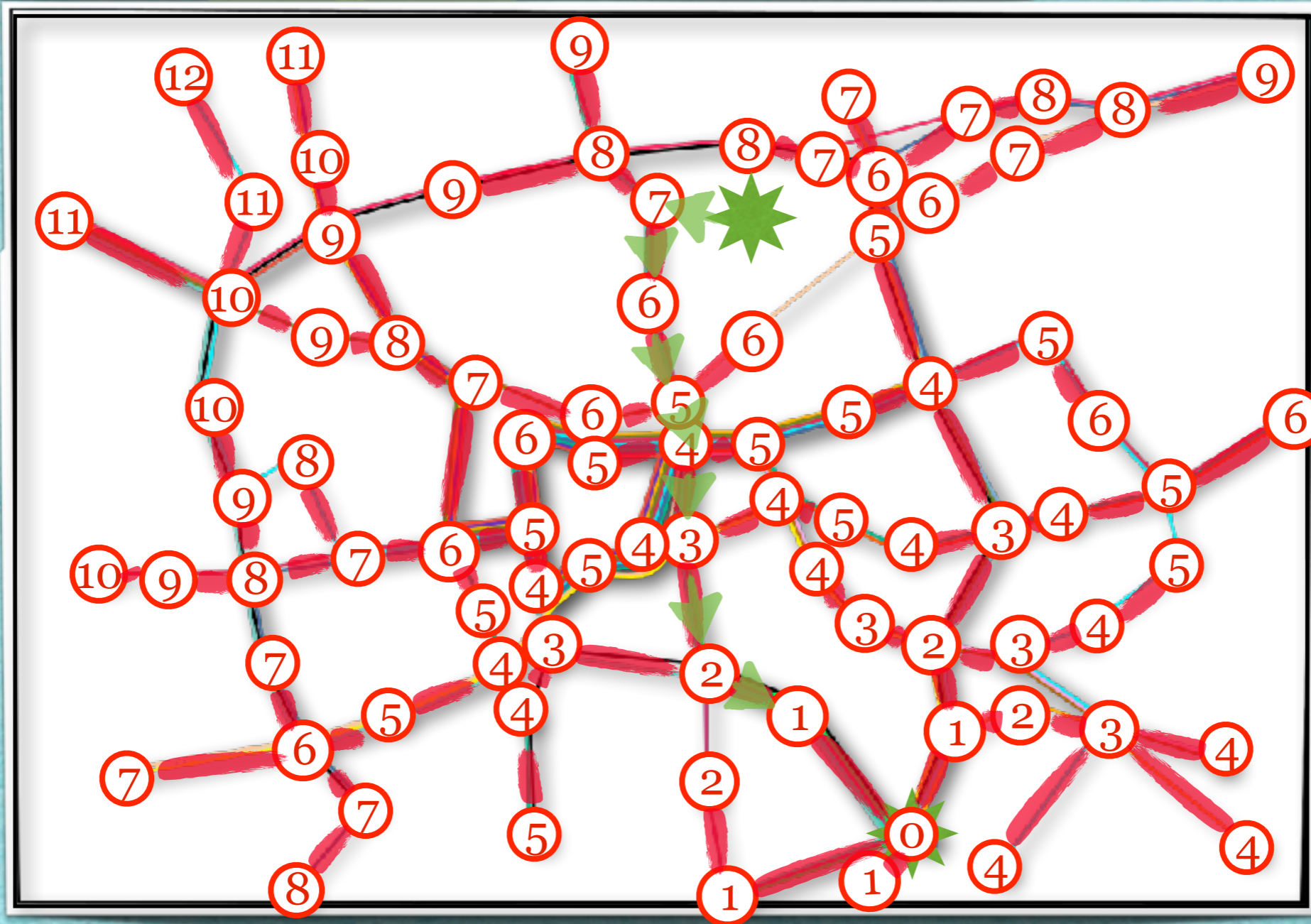
Breitensuche

# Wellenreiten in Graphen



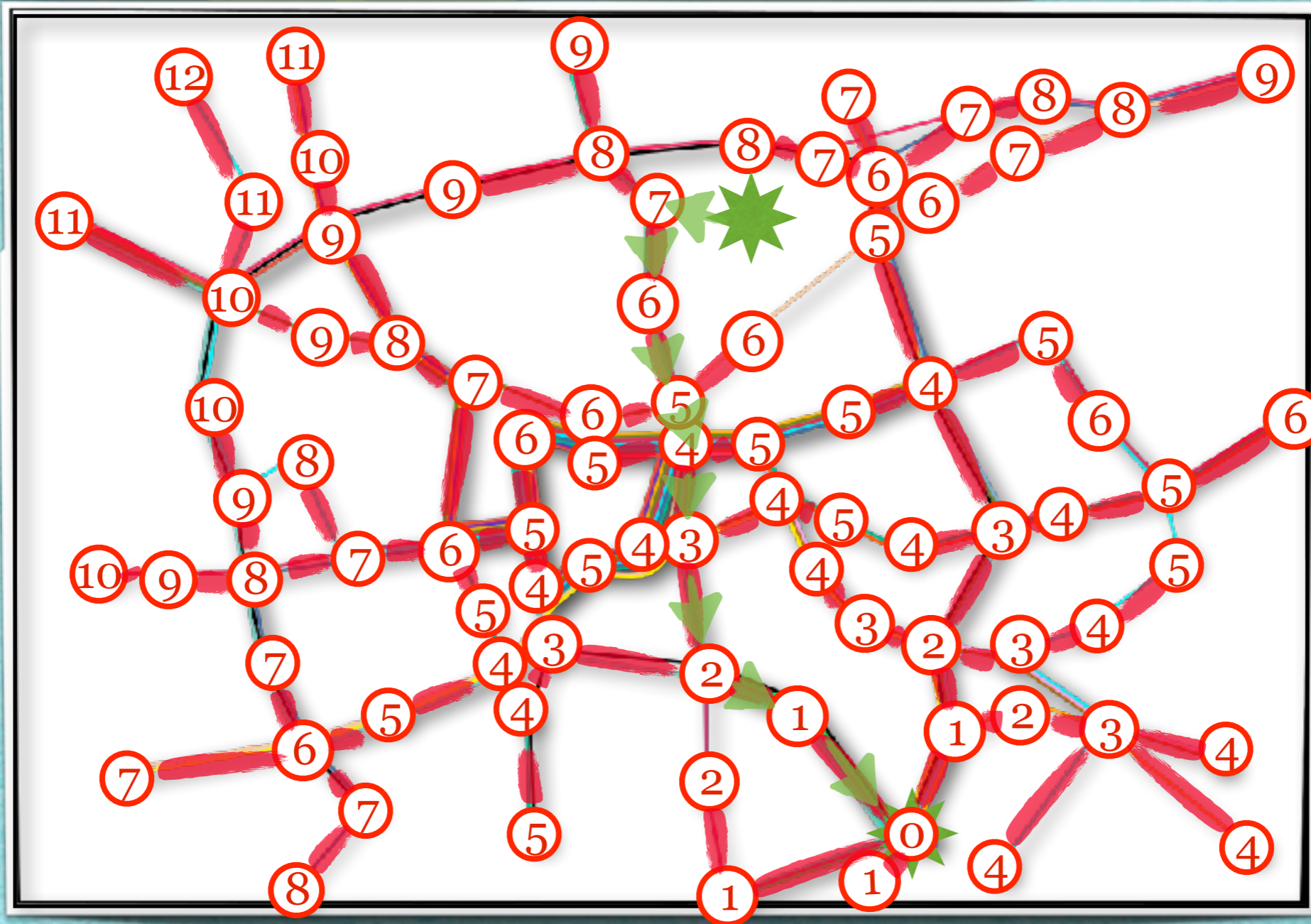
Breitensuche

# Wellenreiten in Graphen



Breitensuche

# Wellenreiten in Graphen



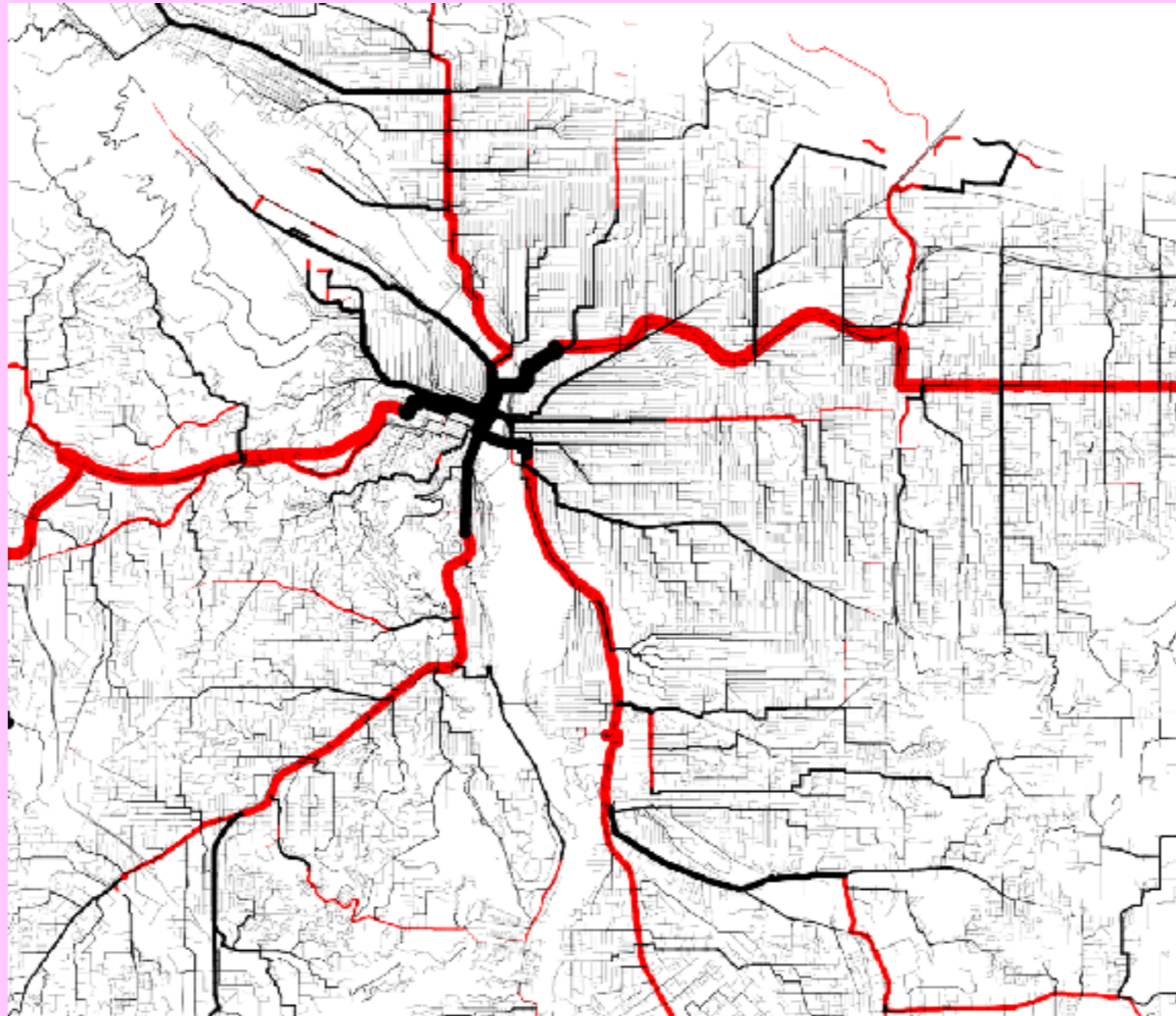
Breitensuche

Breitensuche liefert kürzeste Wege von einer Quelle aus



Viele "Sammler"

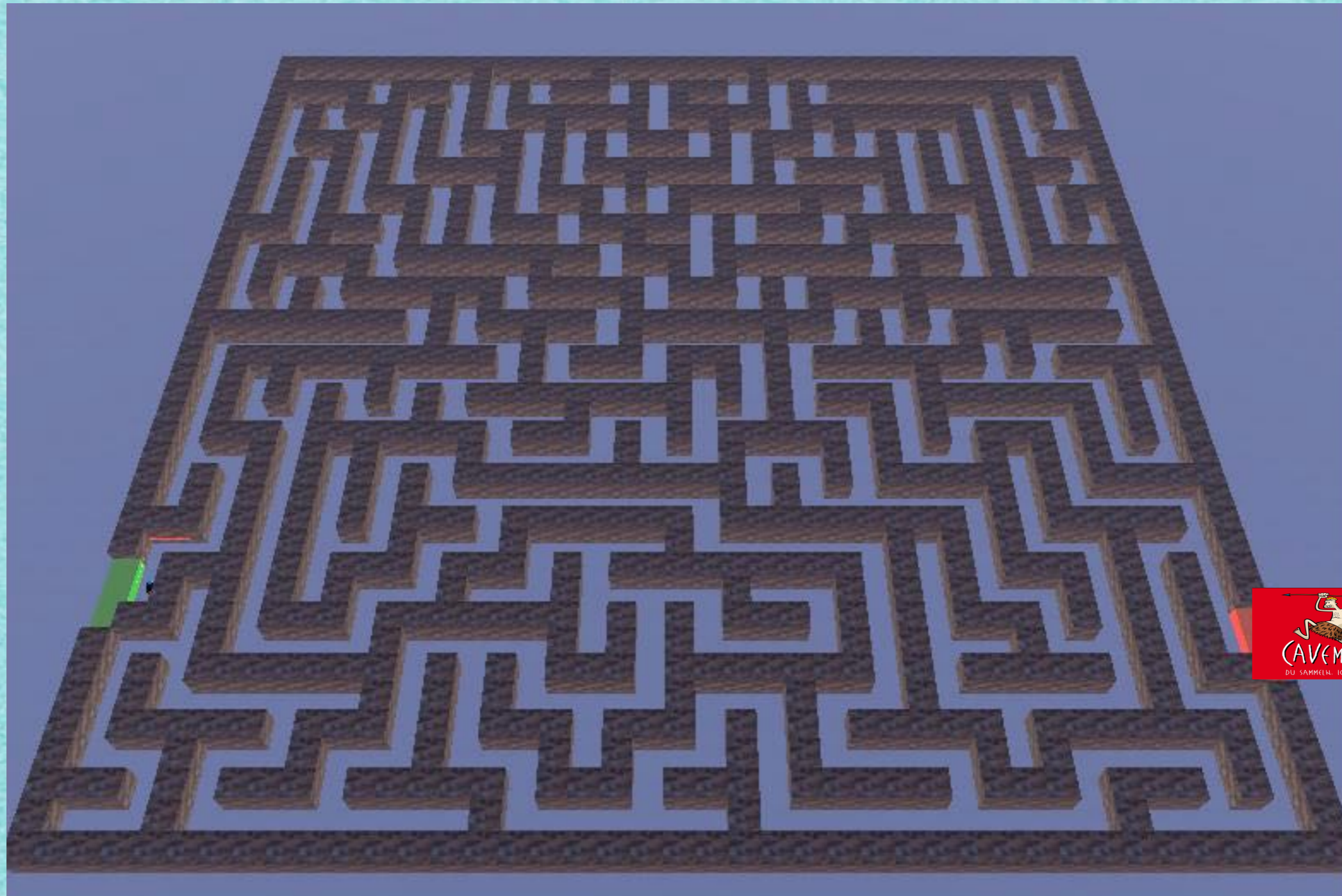
**Breitensuche liefert kürzeste Wege von einer Quelle aus**



**Viele "Sammler"**

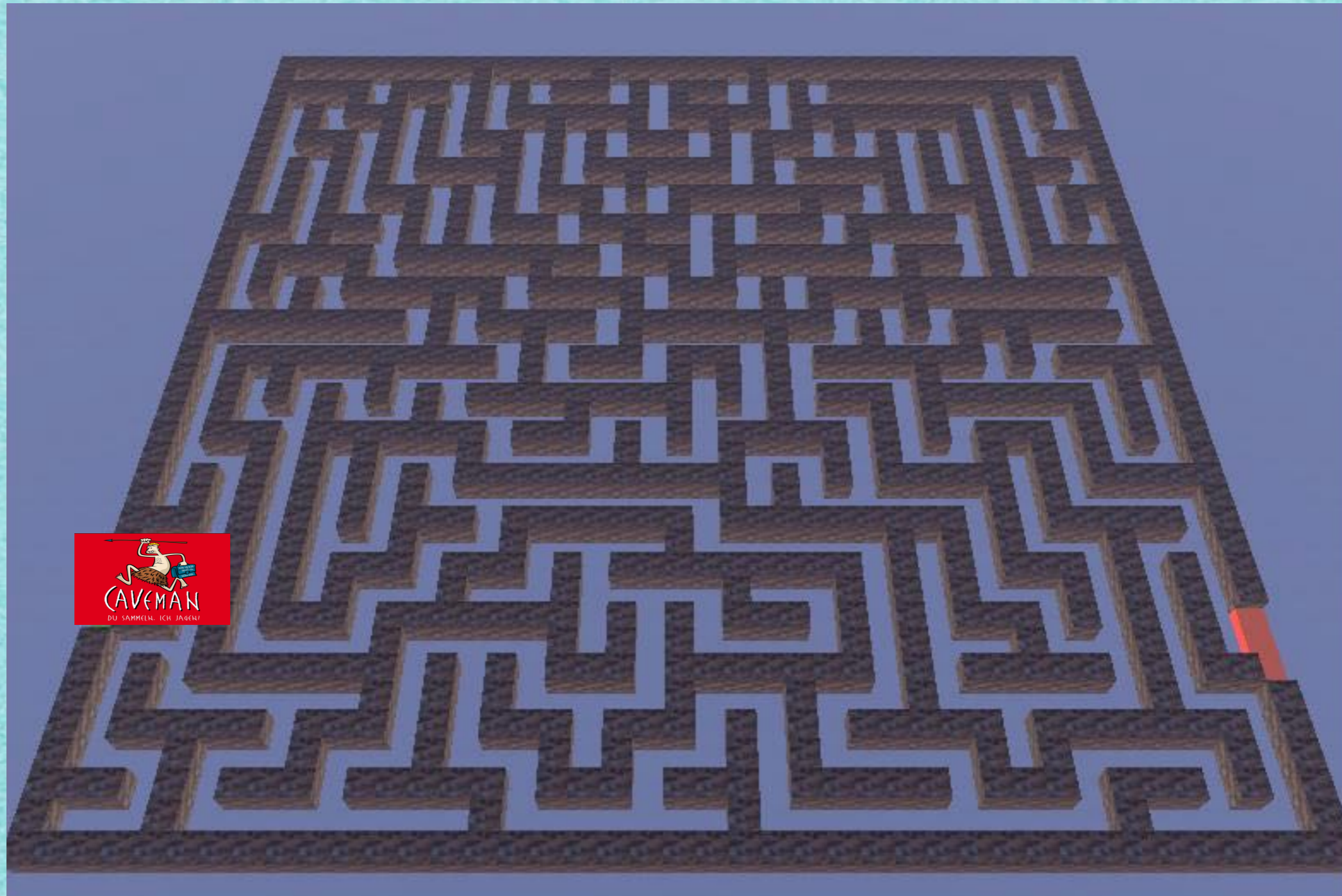


Tiefensuche findet lokal einen Weg durch ein Labyrinth



Ein "Jäger"

## Tiefensuche findet lokal einen Weg durch ein Labyrinth



Ein "Jäger"





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

WENN NAVIGATIONSSYSTEME WIRKLICH WEIBLICH WÄREN...

NA GUT, DANN KANN ICH HALT KEINE KARTEN LEGEN!  
ABER WER WOLLTE DENN NICHT ANHALTEN UND NACH DEM WEG FRAGEN?!



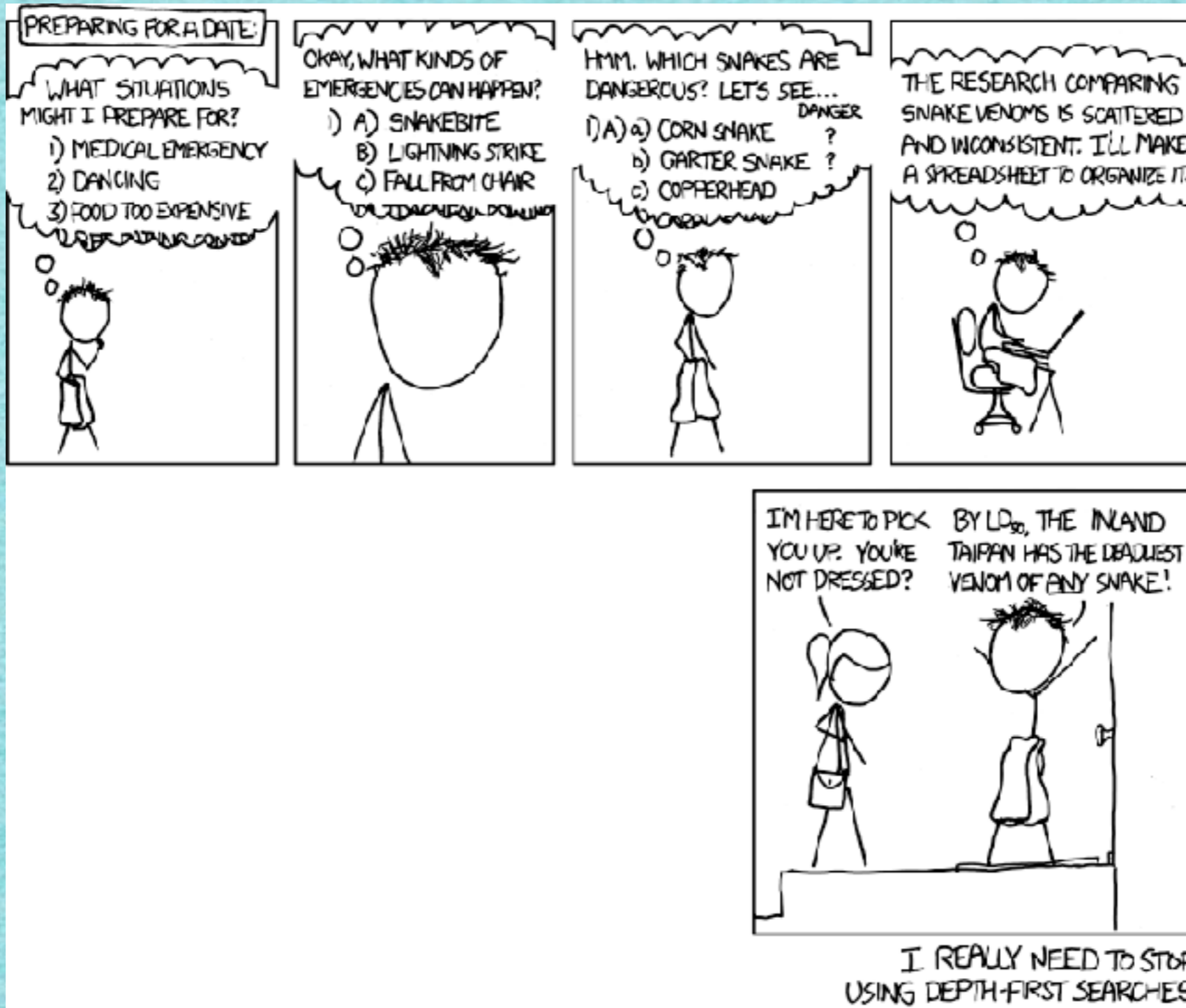
MYCARTOONS.DE



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



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



*Quiz!*

*s.fekete@tu-bs.de*