

Kapitel 3.8: Laufzeit von DFS und BFS

*Algorithmen und Datenstrukturen
WS 2023/24*

Prof. Dr. Sándor Fekete

SATZ 3.13

Der Graphen-Scan-Algorithmus 3.7 lässt sich so implementieren, dass die Laufzeit $O(n+m)$ ist.

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 $w \in$

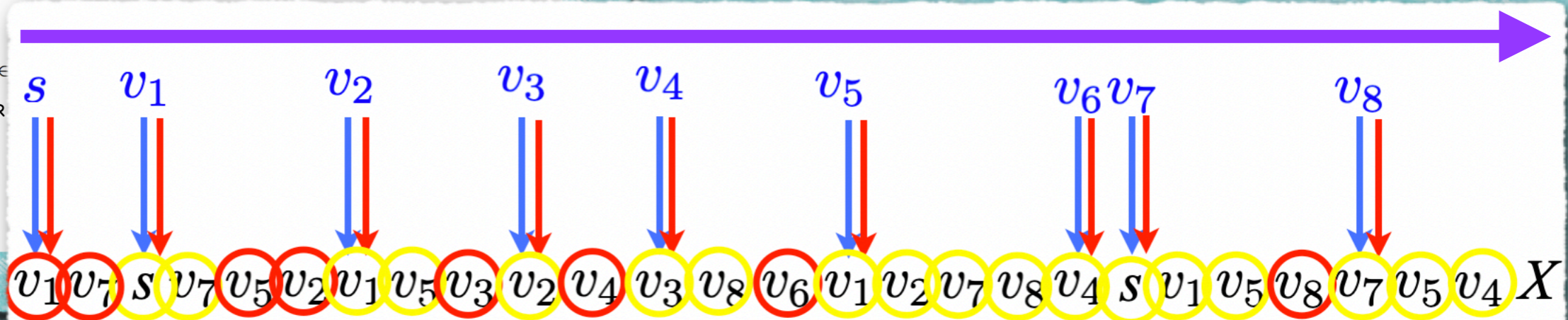
2.3.2. Setze $R := R \cup \{w\}$

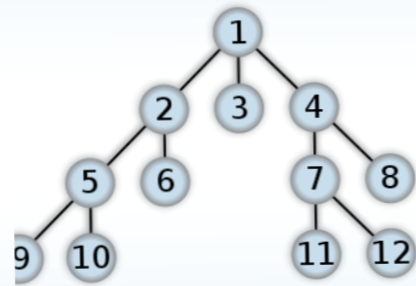
}

}

3. STOP

Adjazenzliste!





Kapitel 3.9: Eigenschaften von DFS und BFS

*Algorithmen und Datenstrukturen
WS 2023/24*

Prof. Dr. Sándor Fekete

Algorithmus 3.17

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

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

für jeden Knoten $v \in Y$ die Länge $l(v)$ eines kürzesten s - v -Weges,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei $R := \{s\}$, $Y := \{s\}$, $T := \emptyset$, $l(s) := 0$
2. WHILE ($R \neq \emptyset$) DO {
 - 2.1. wähle Element $v \in R$
 - 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 $w \in V \setminus R$ mit $e = \{v, w\} \in E$;
 - 2.3.2. setze $R := R \cup \{w\}$, $Y := Y \cup \{w\}$, $T := T \cup \{e\}$;
 - 2.3.3. setze $l(w) := l(v) + 1$}

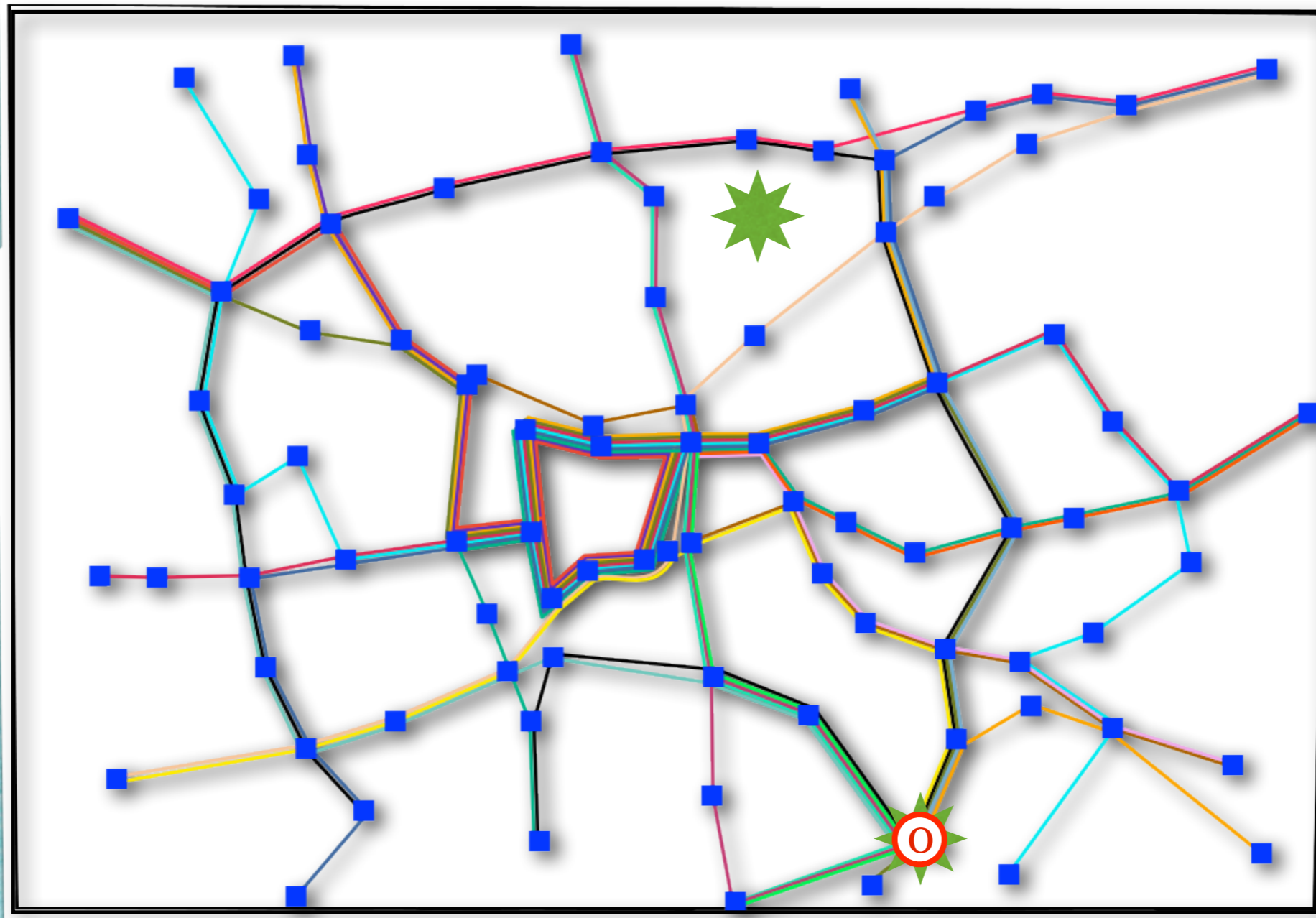
Satz 3.18

- (1) *Das Verfahren 3.17 ist endlich.*
- (2) *Die Laufzeit ist $O(n+m)$.*
- (3) *Am Ende ist für jeden erreichbaren Knoten $v \in Y$ die Länge eines kürzesten Weges von s nach v **im Baum (Y,T)** durch $l(v)$ gegeben.*
- (4) *Am Ende ist für jeden erreichbaren Knoten $v \in Y$ die Länge eines kürzesten Weges von s nach v **im Graphen (V,E)** durch $l(v)$ gegeben.*

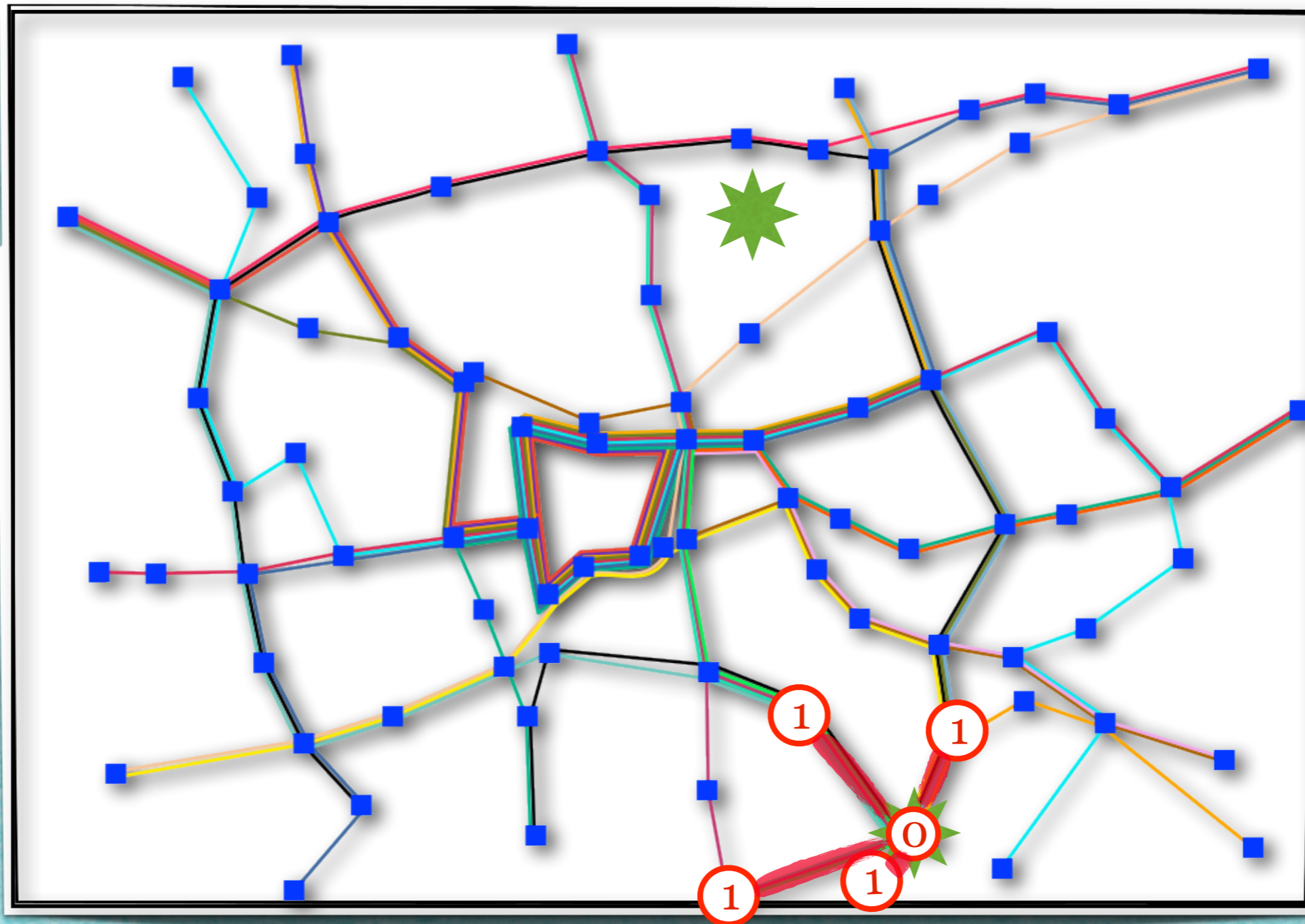
Beweis:

- (1) **Wie für Algorithmus 3.7 gelten alle Eigenschaften. zusätzlich ist für jeden Knoten $v \in Y$ per Induktion, der Wert $l(v)$ tatsächlich definiert.**
- (2) **Die Laufzeit bleibt von Algorithmus 3.7 erhalten.**

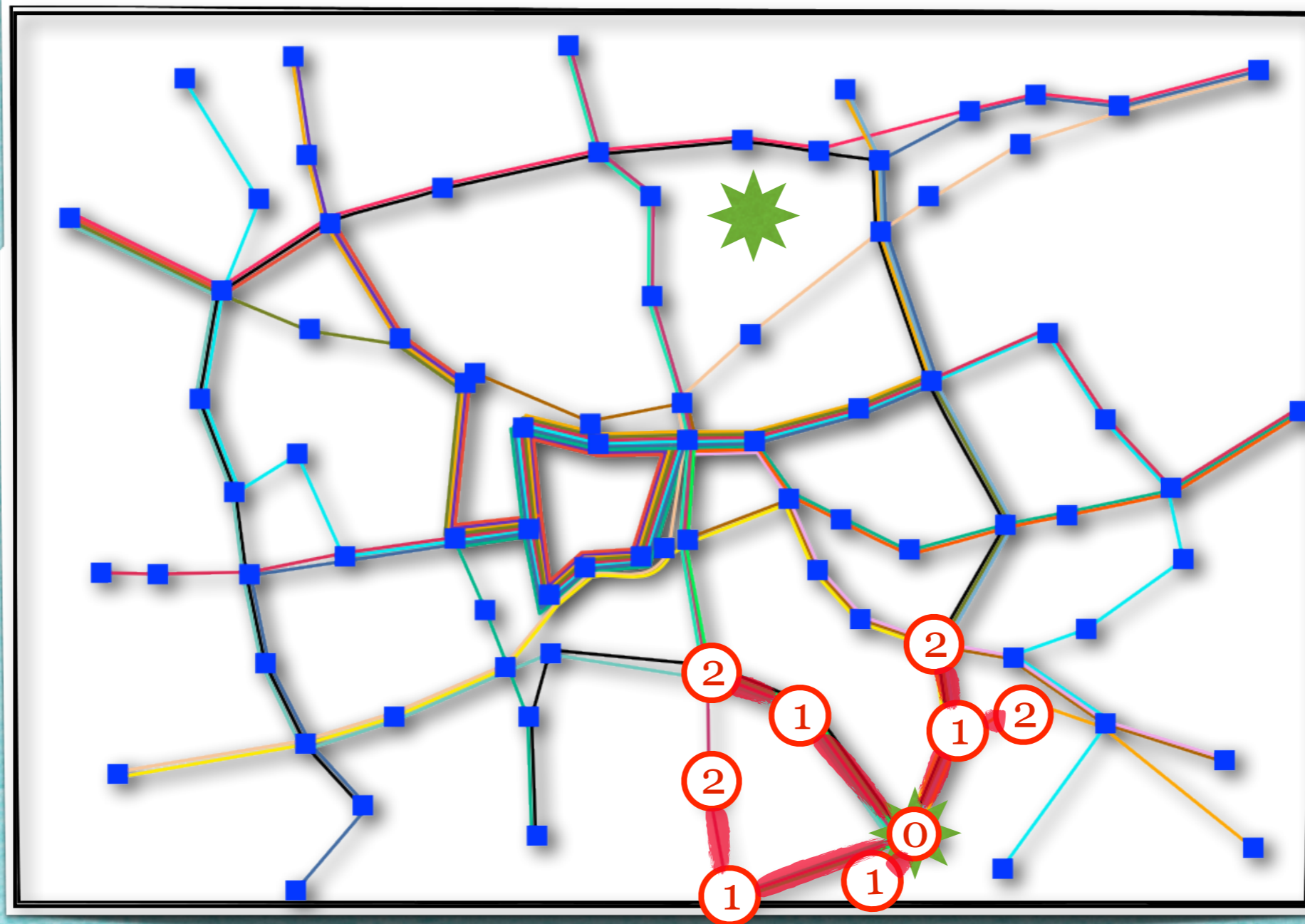
Wellenreiten in Graphen



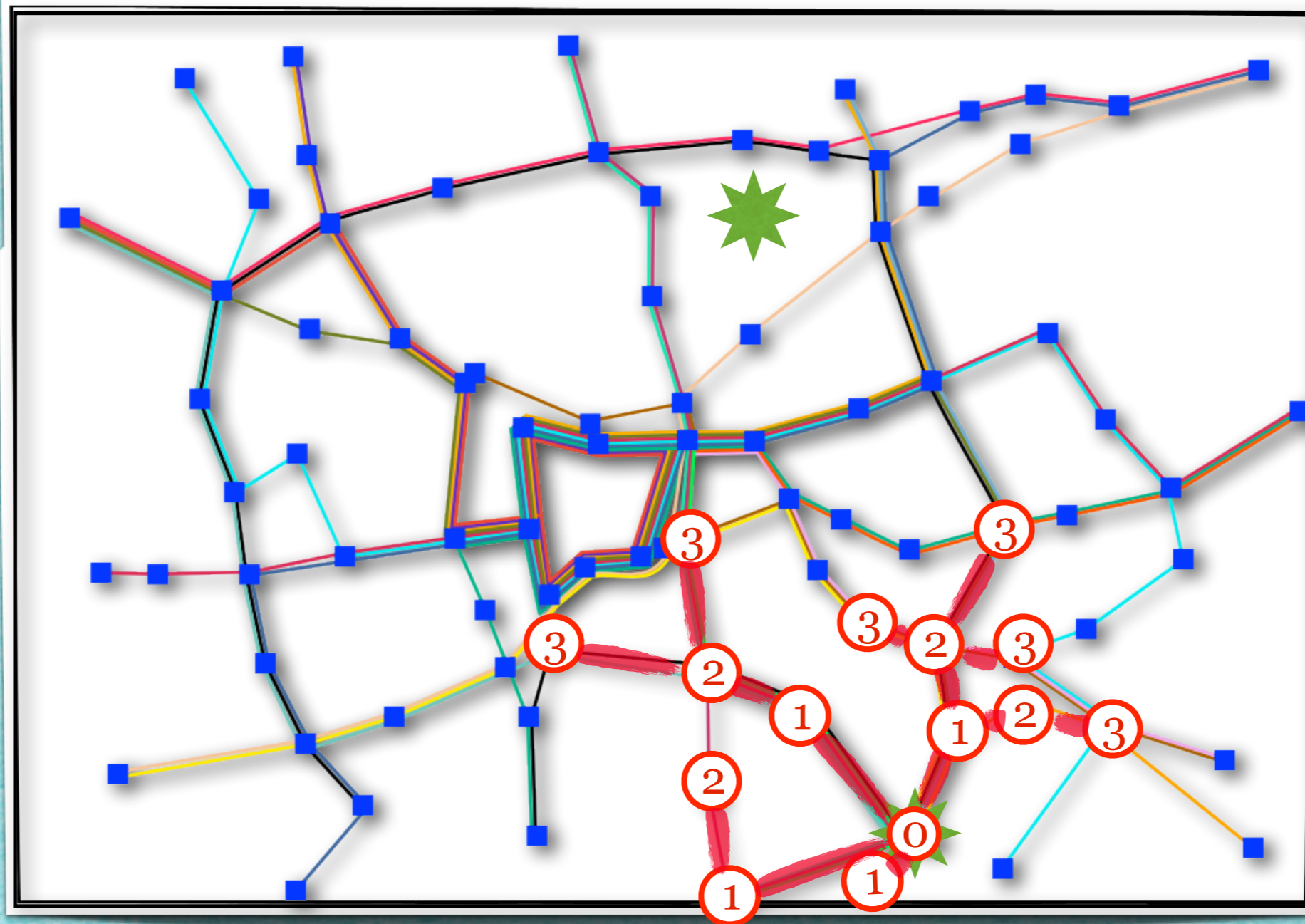
Wellenreiten in Graphen



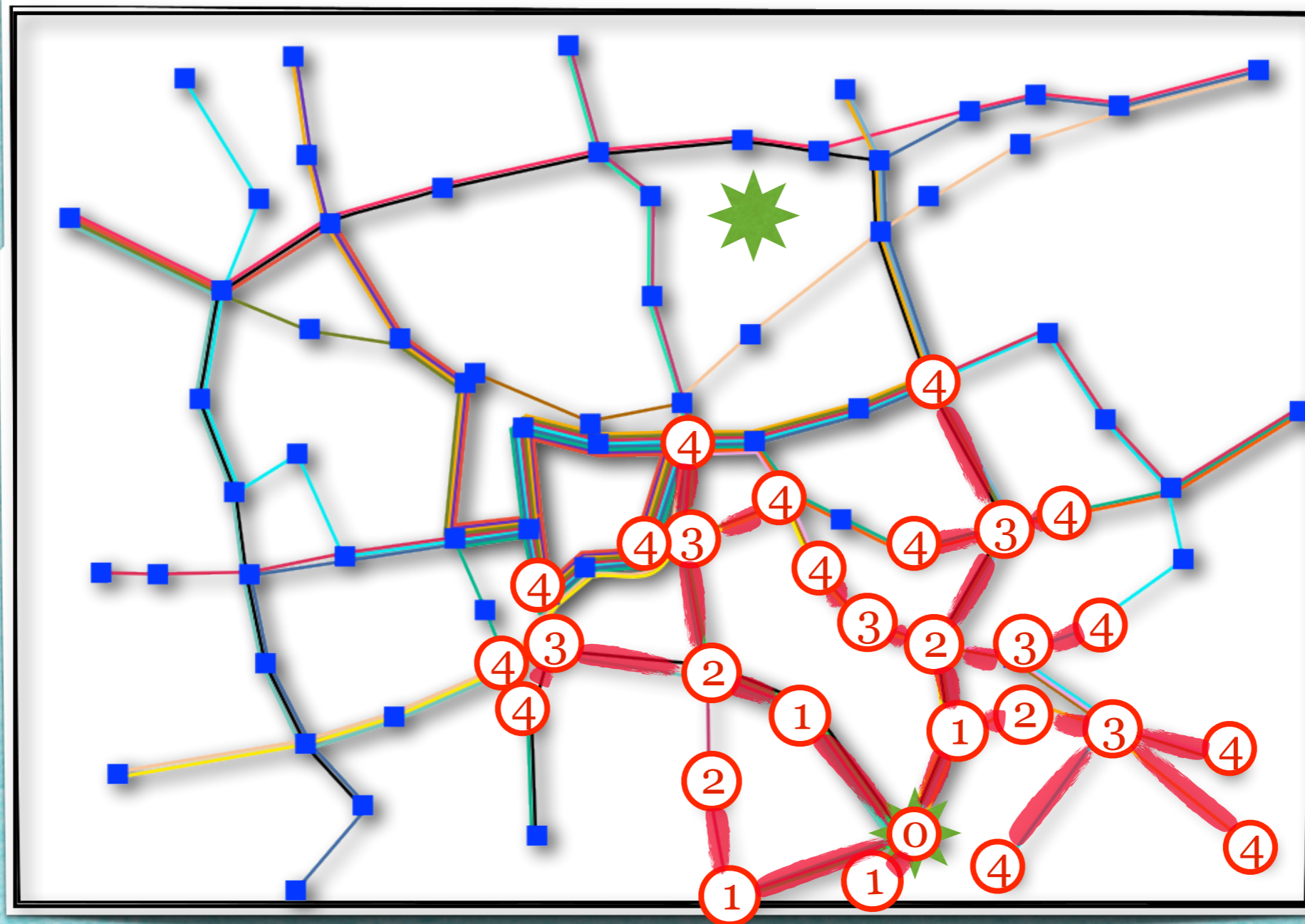
Wellenreiten in Graphen



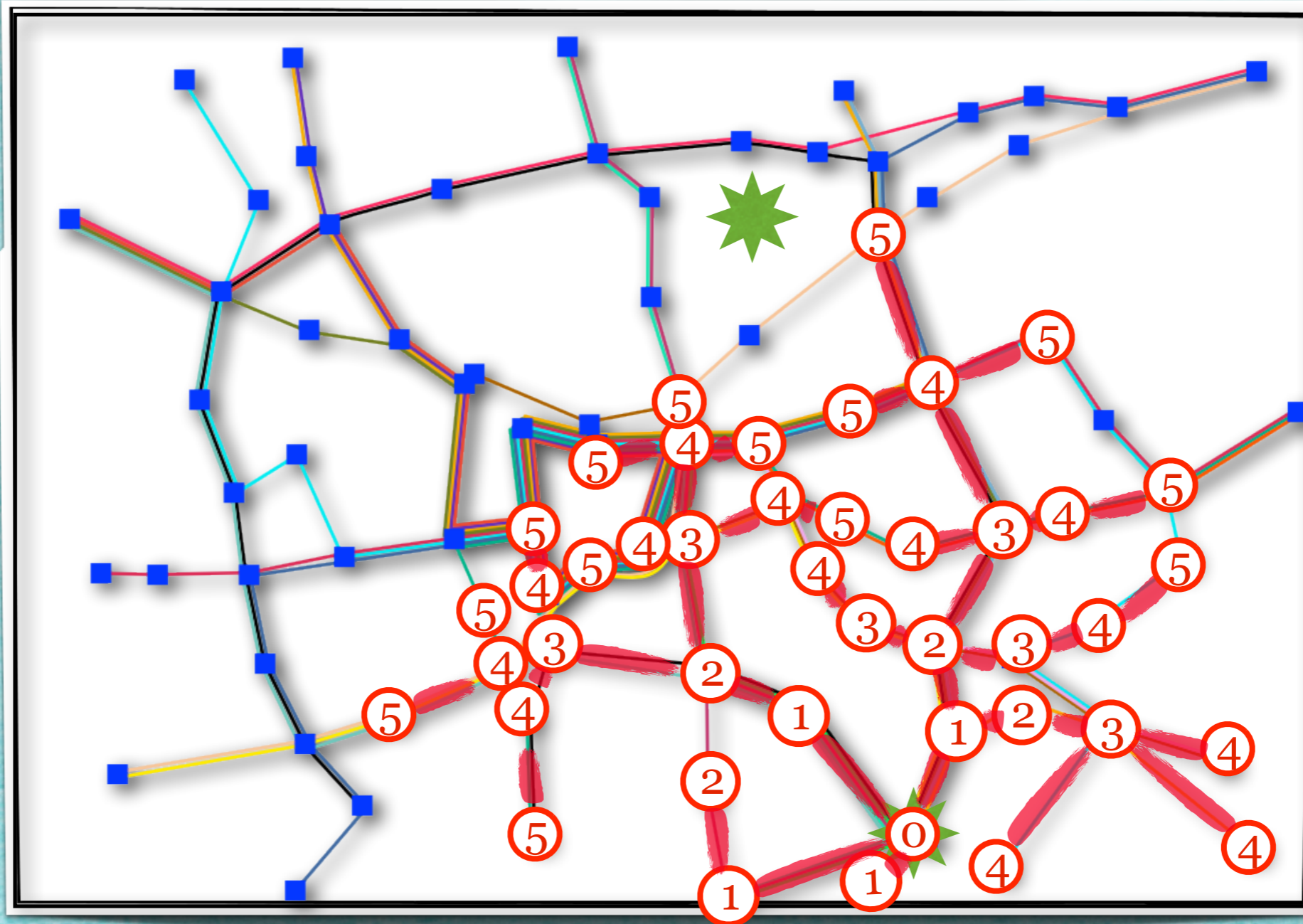
Wellenreiten in Graphen



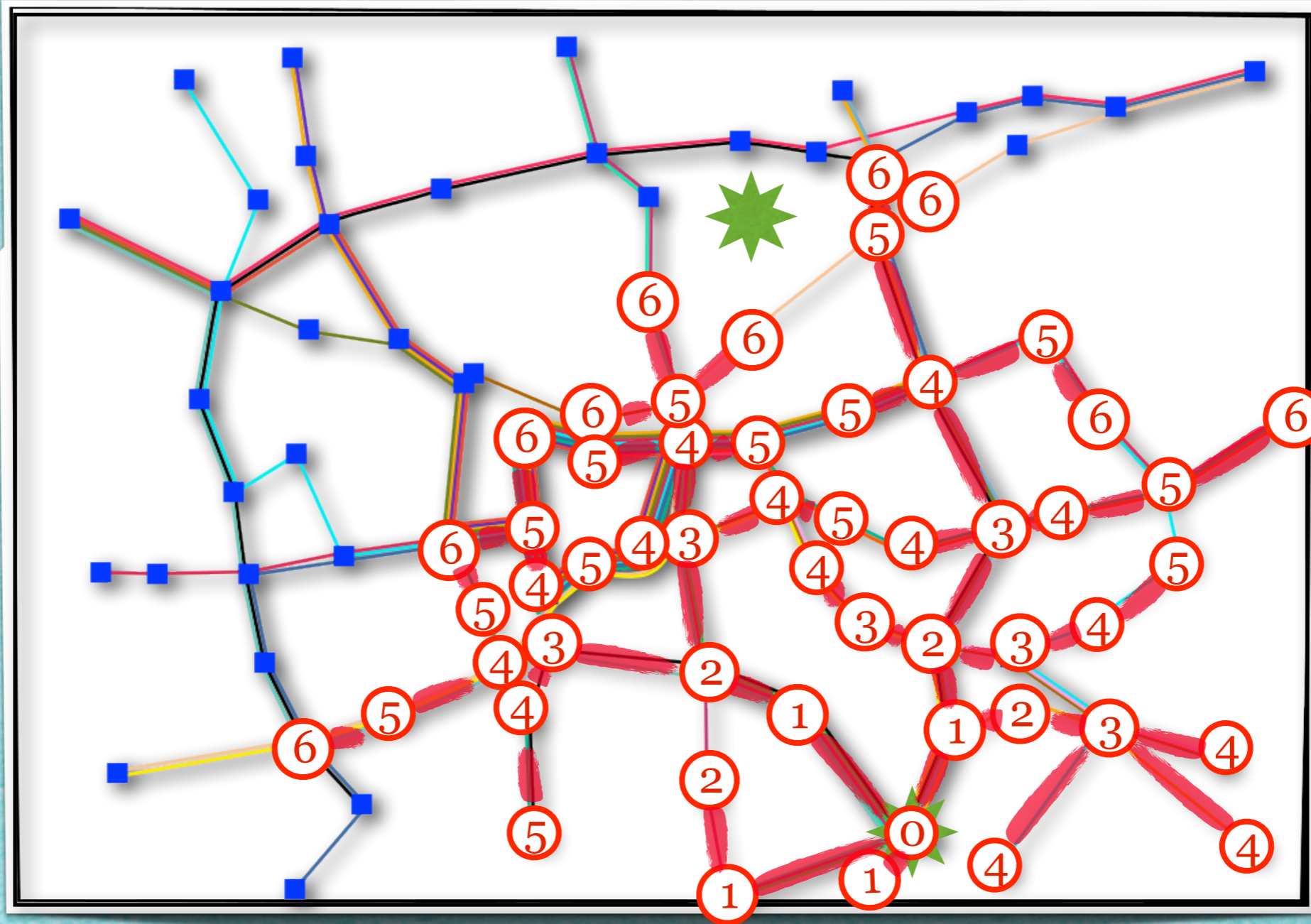
Wellenreiten in Graphen



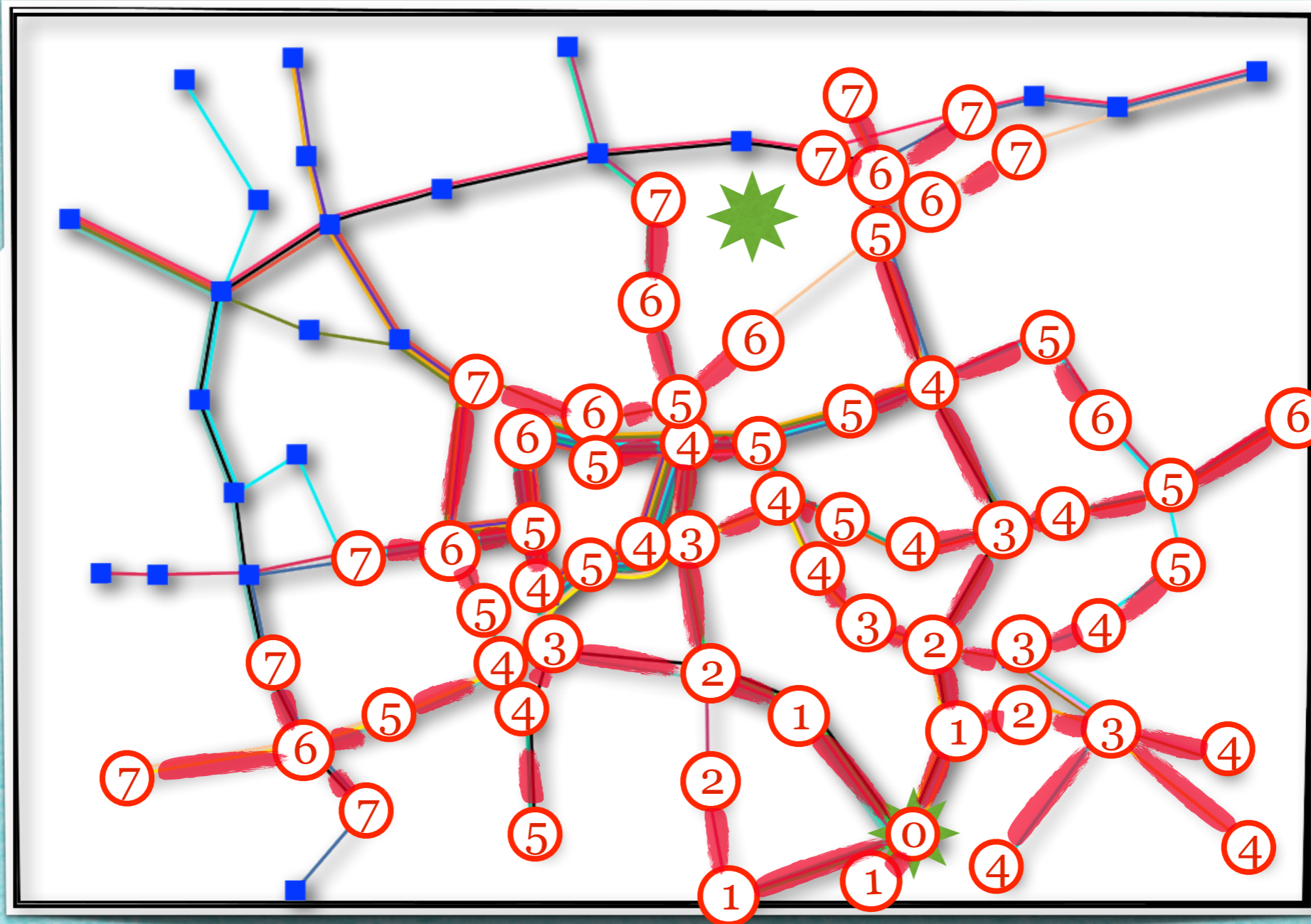
Wellenreiten in Graphen



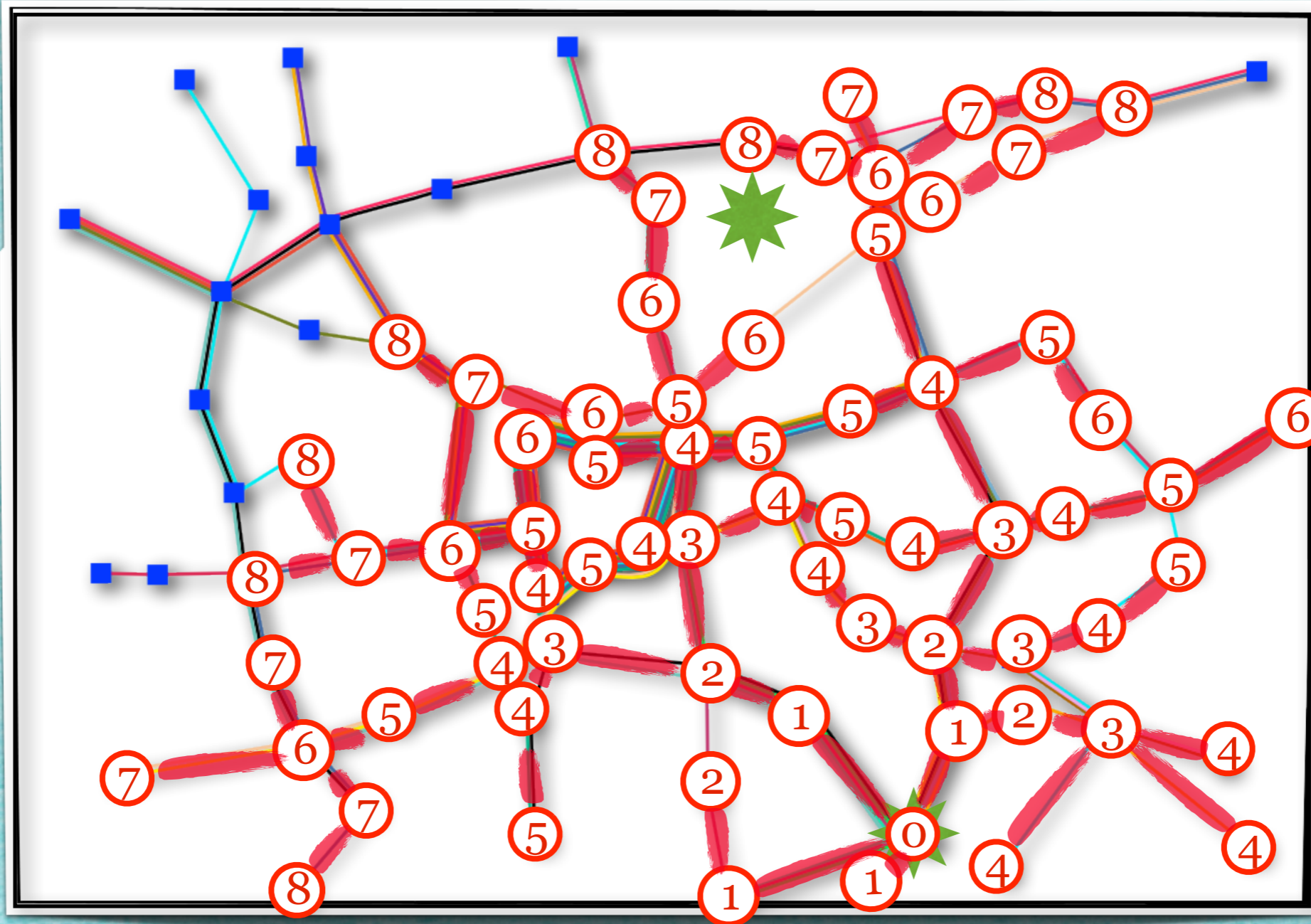
Wellenreiten in Graphen



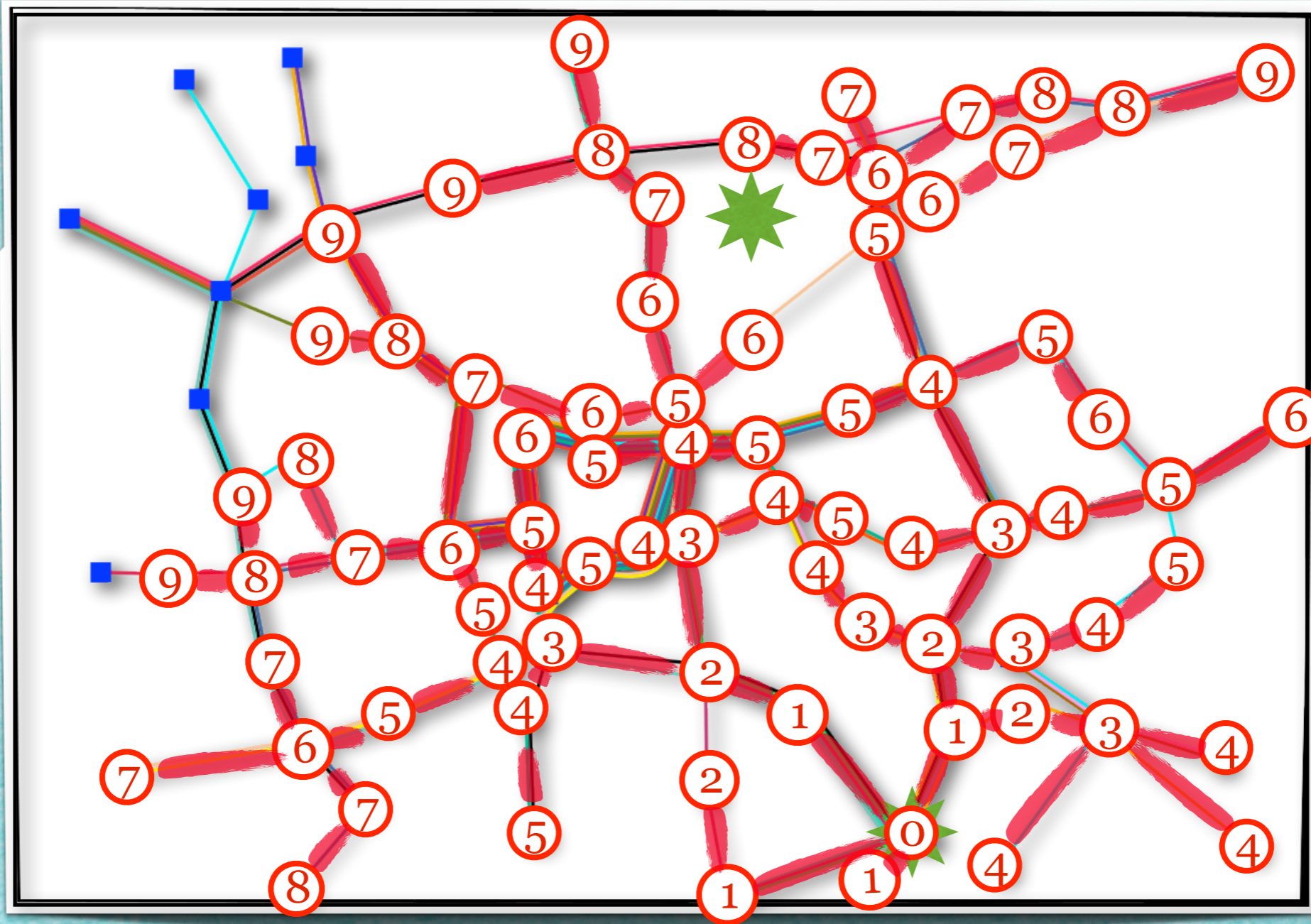
Wellenreiten in Graphen



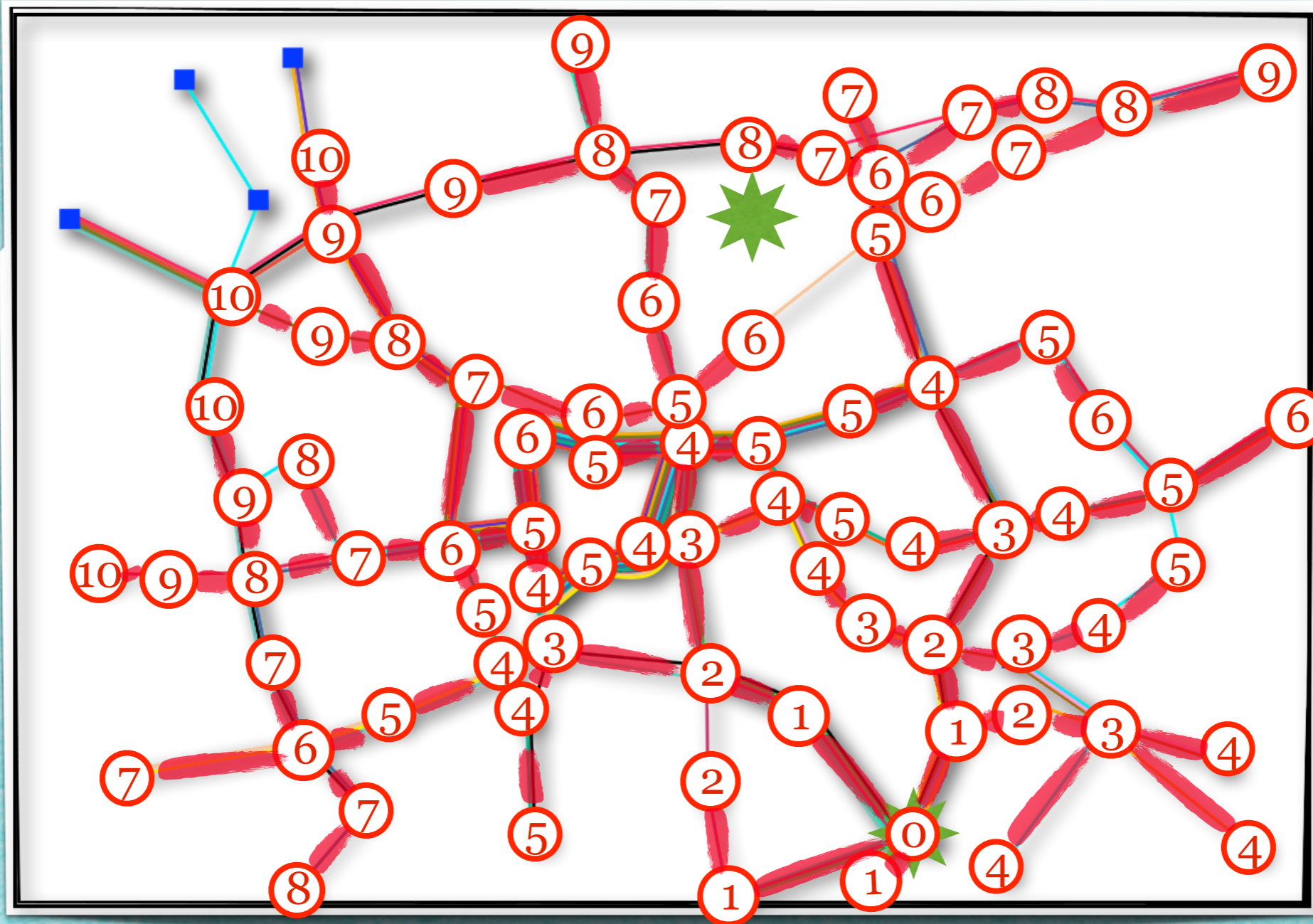
Wellenreiten in Graphen



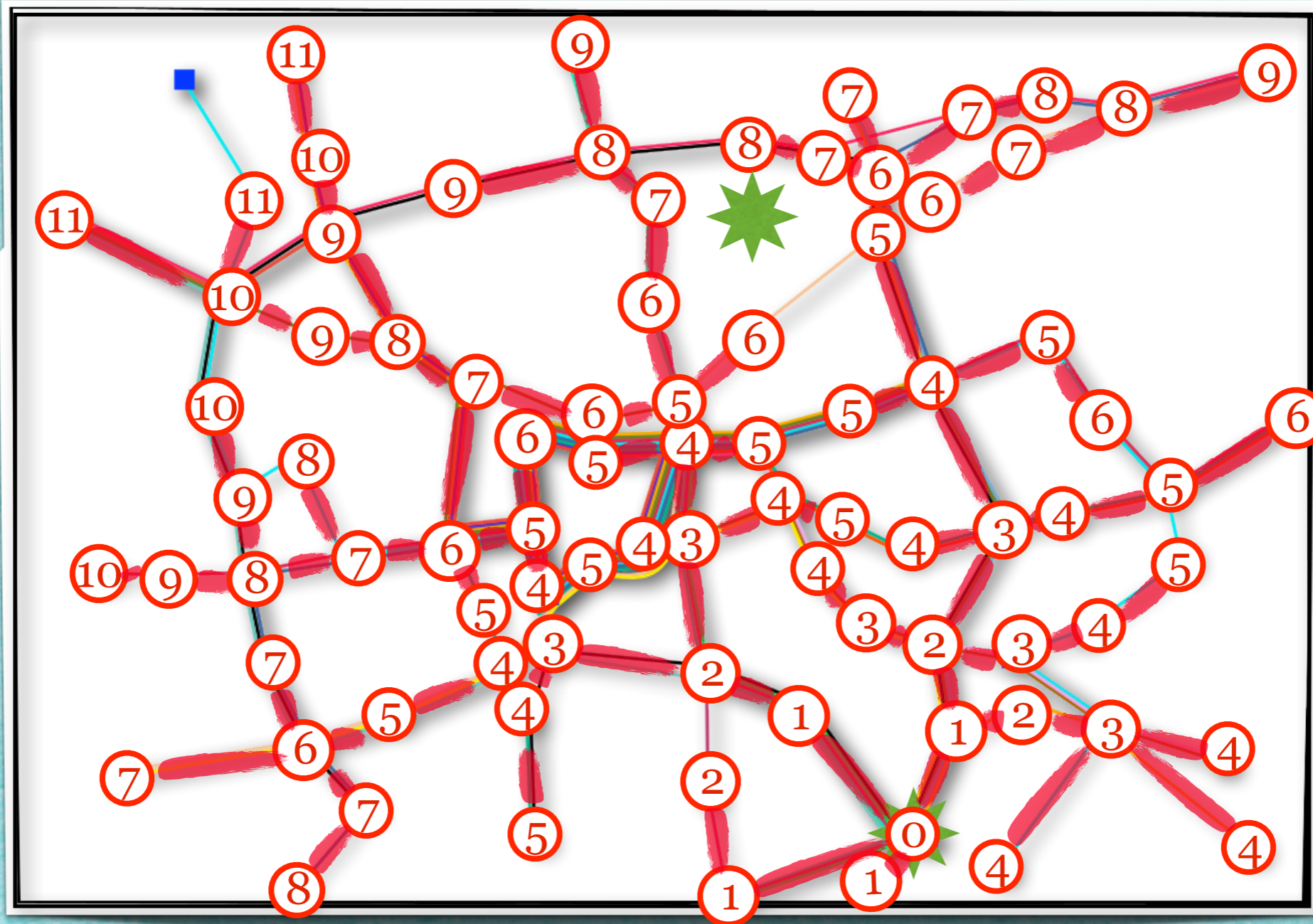
Wellenreiten in Graphen



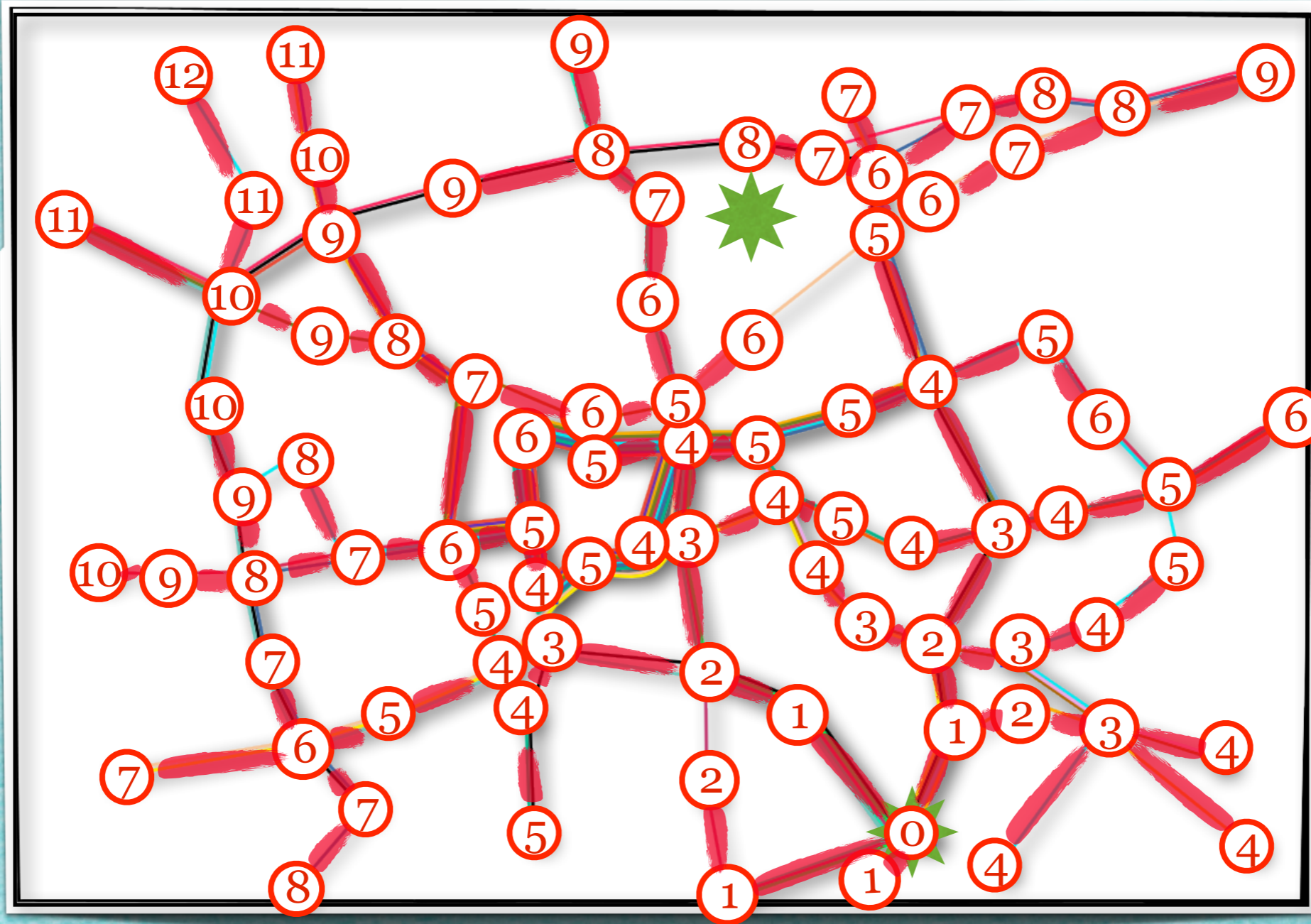
Wellenreiten in Graphen



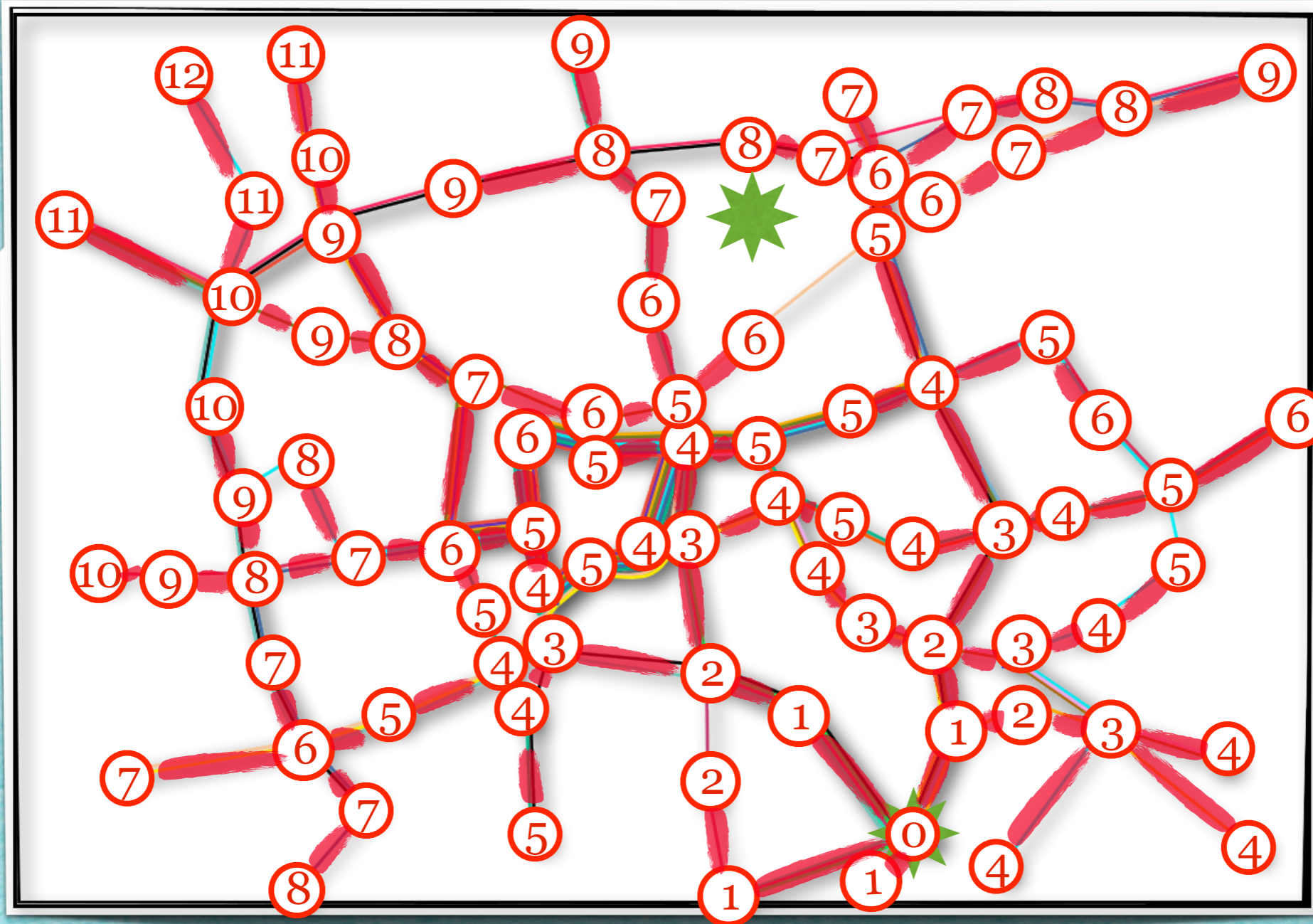
Wellenreiten in Graphen



Wellenreiten in Graphen

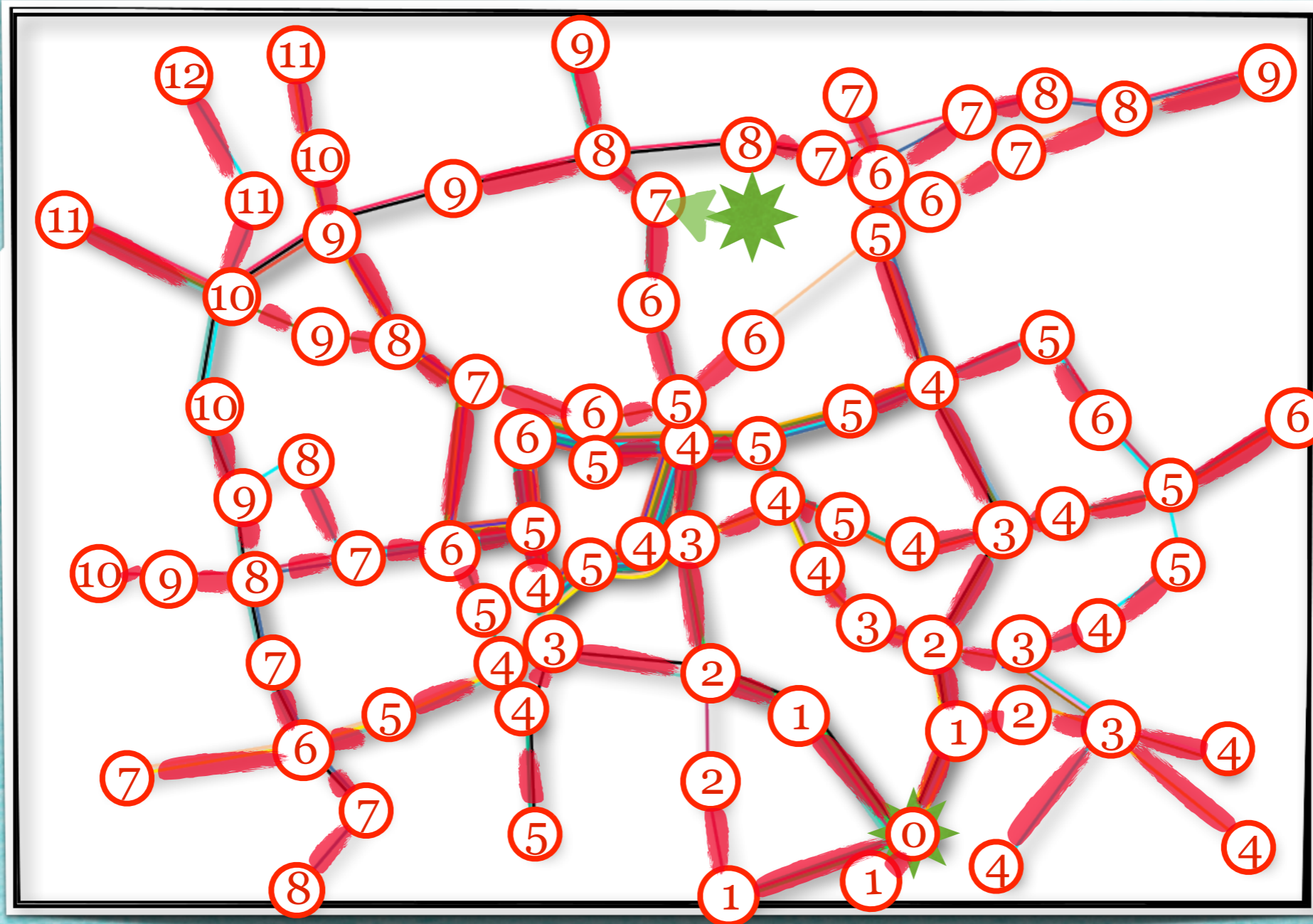


Wellenreiten in Graphen



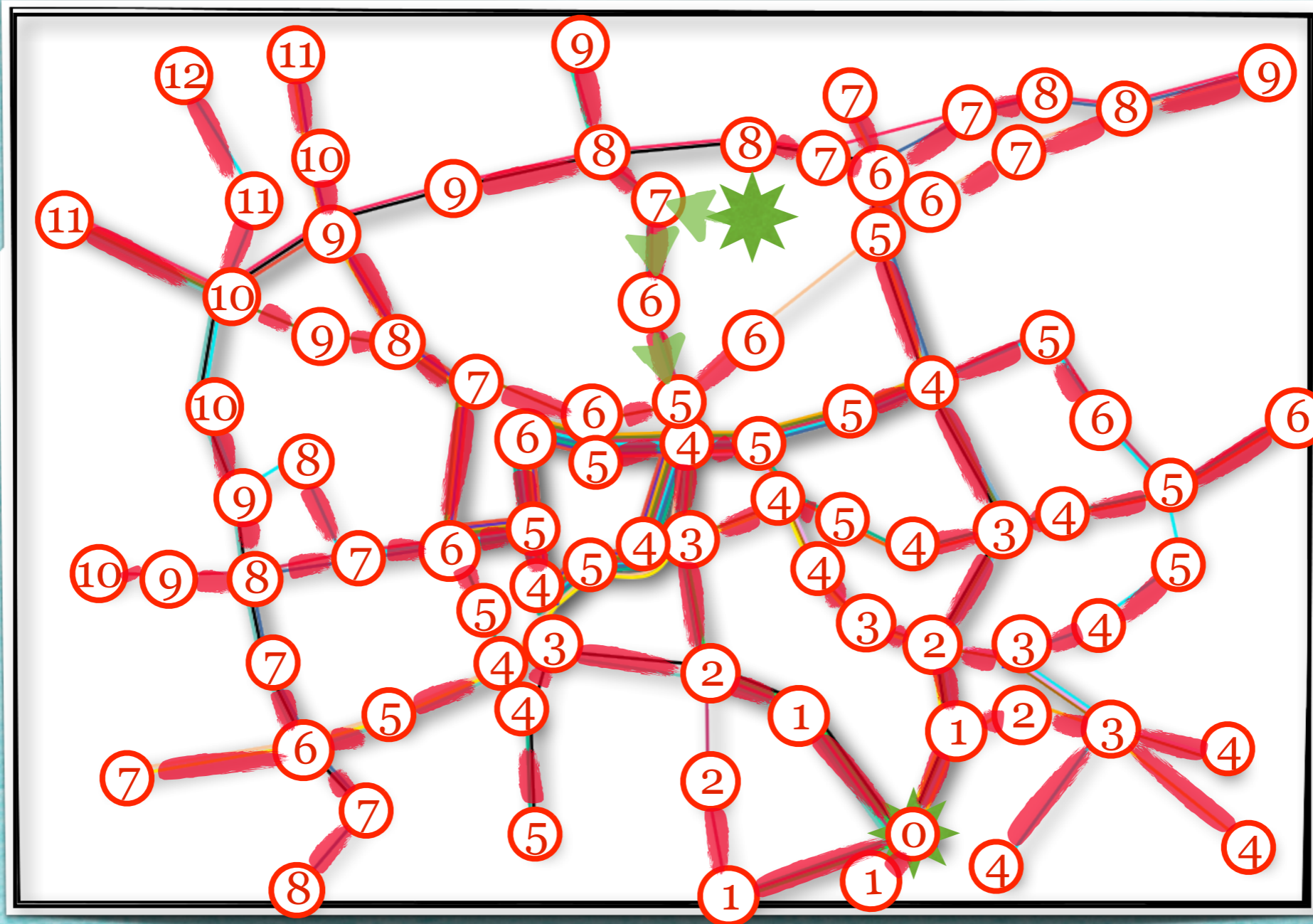
Breitensuche

Wellenreiten in Graphen



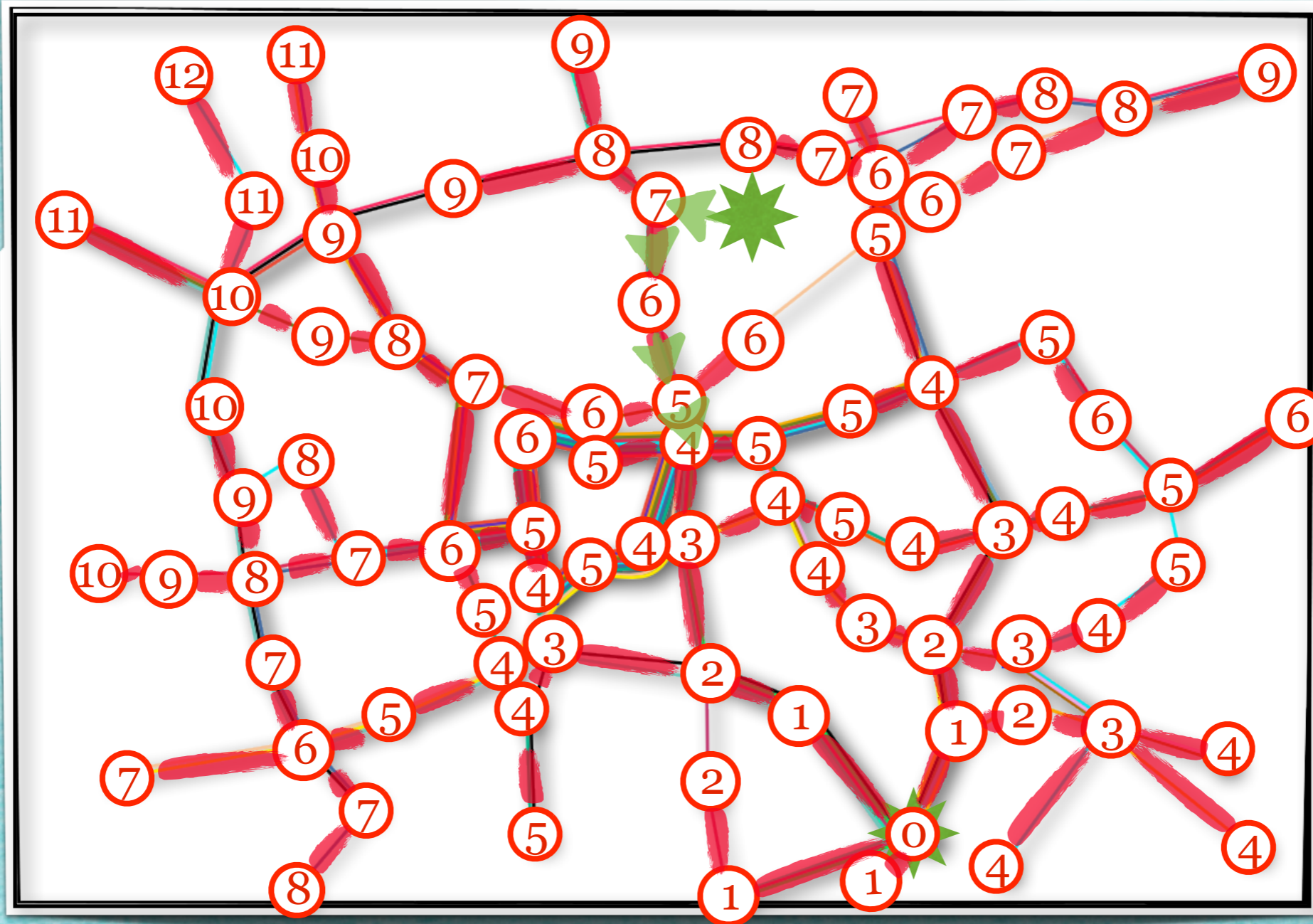
Breitensuche

Wellenreiten in Graphen



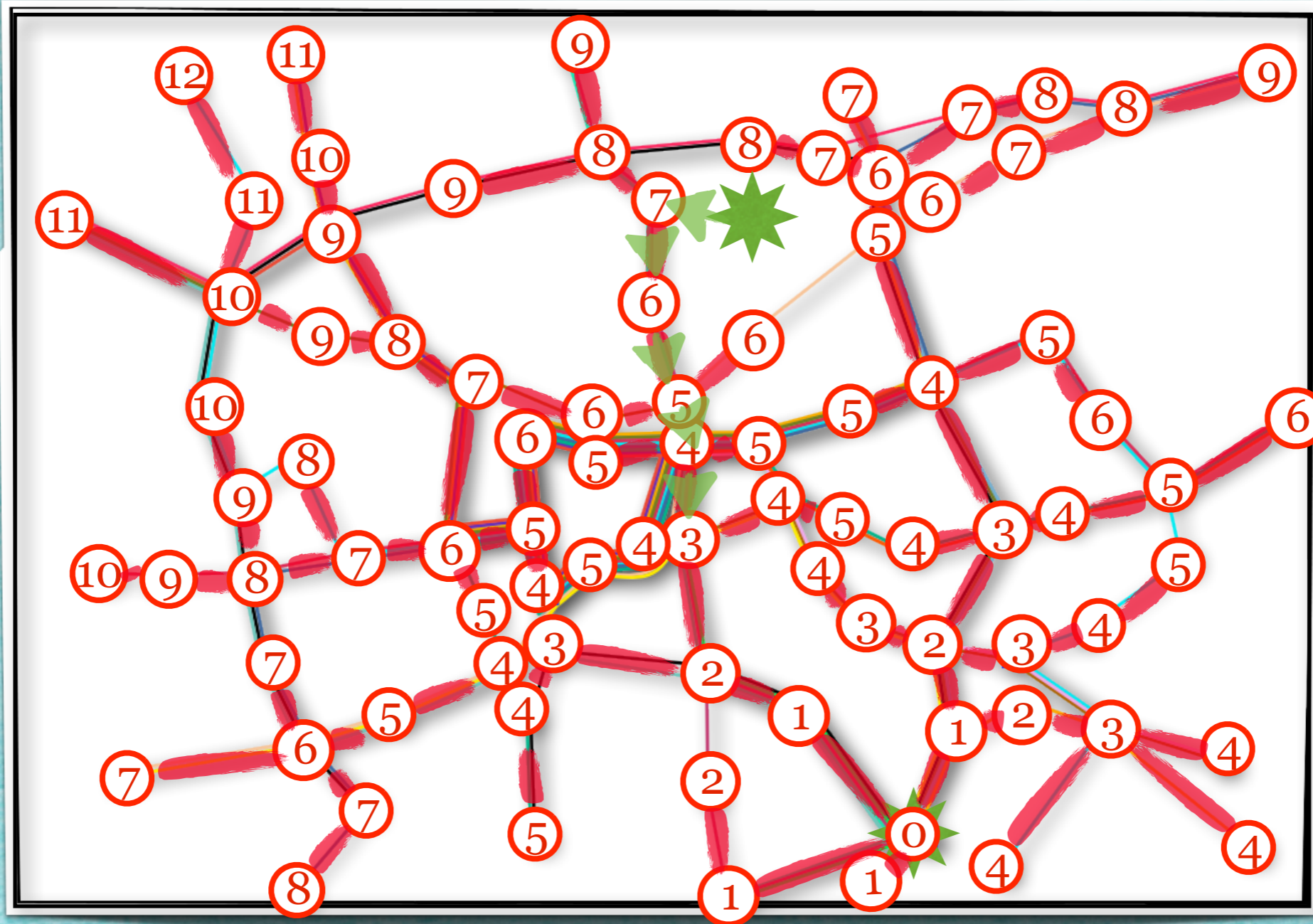
Breitensuche

Wellenreiten in Graphen



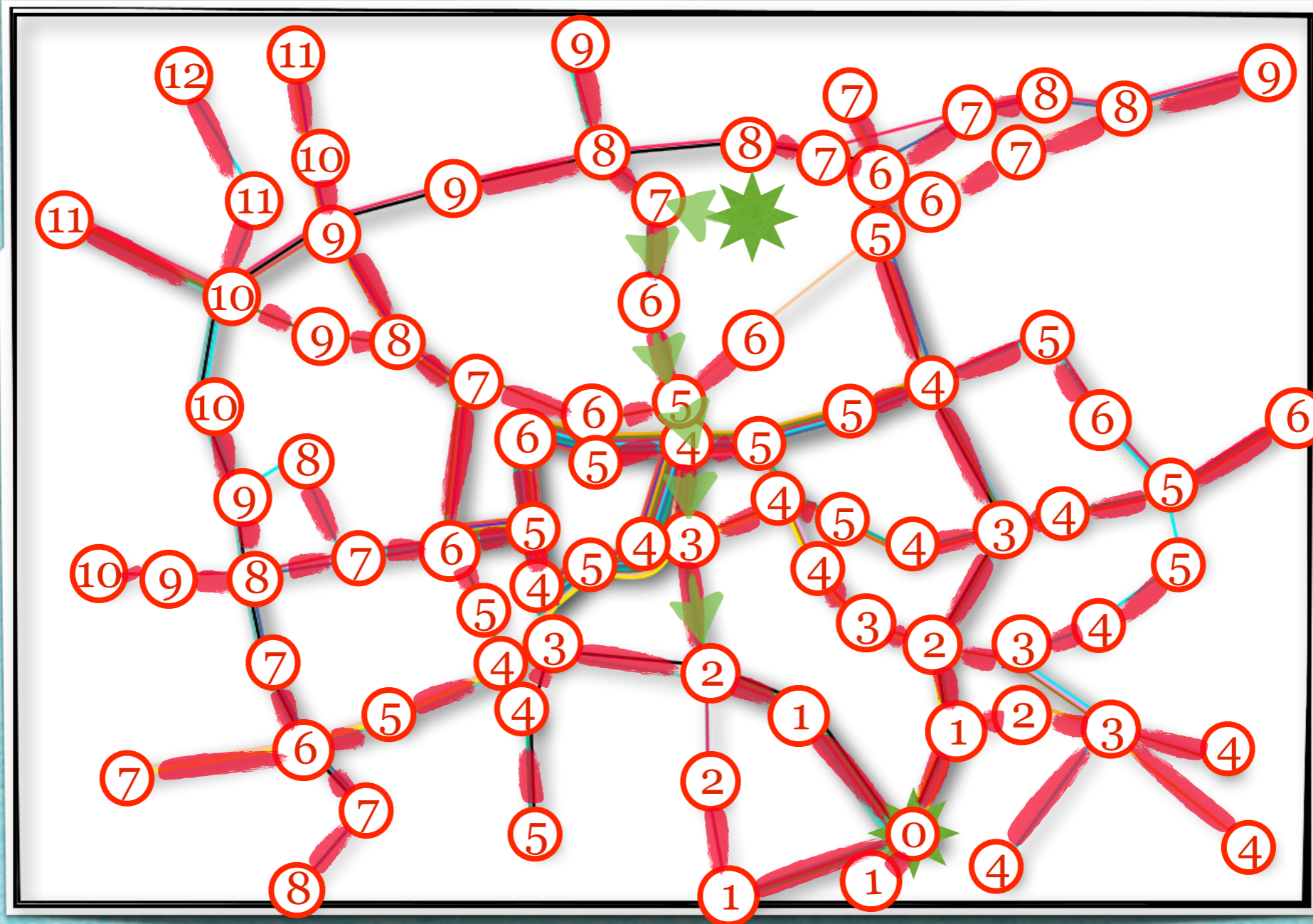
Breitensuche

Wellenreiten in Graphen



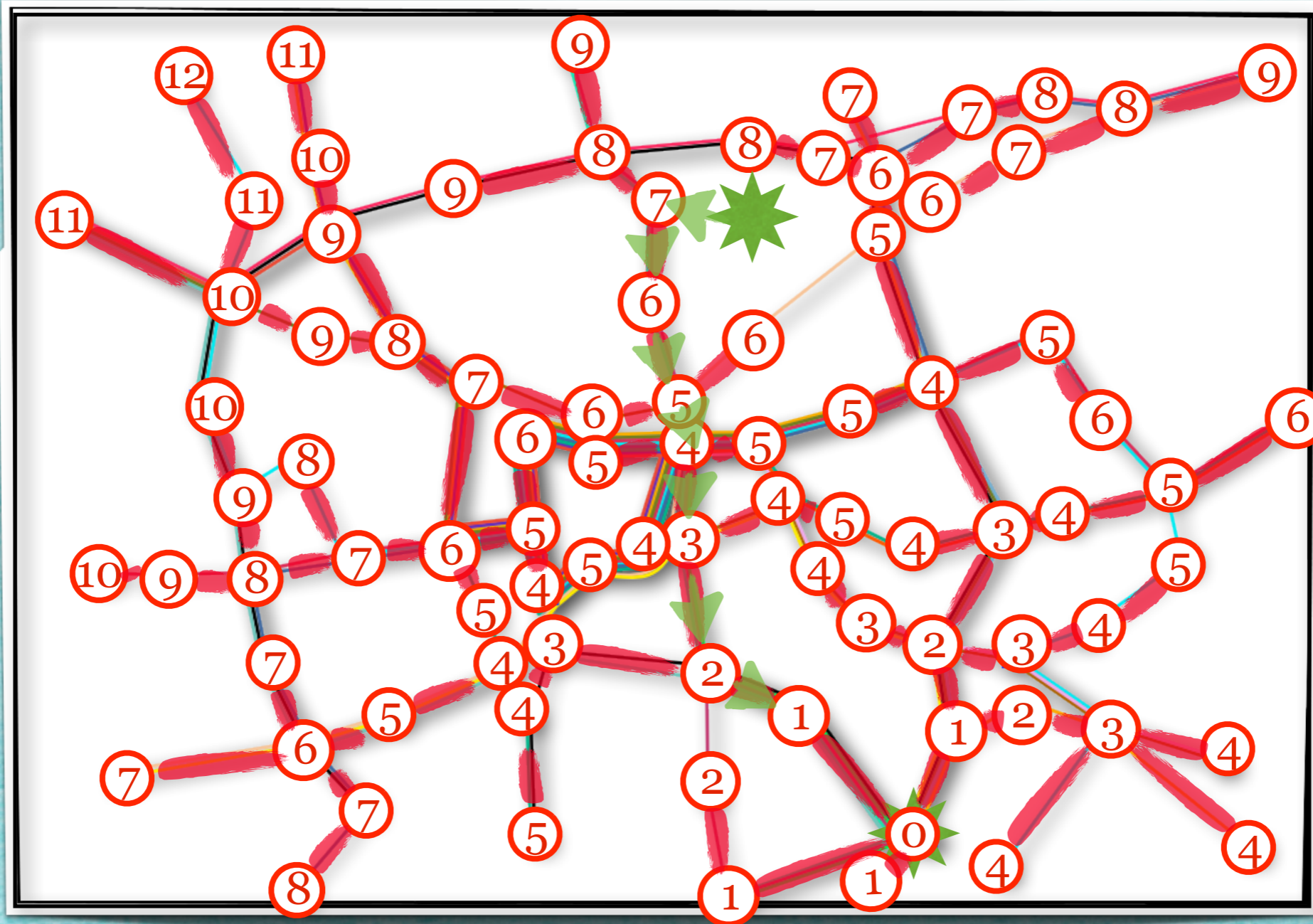
Breitensuche

Wellenreiten in Graphen



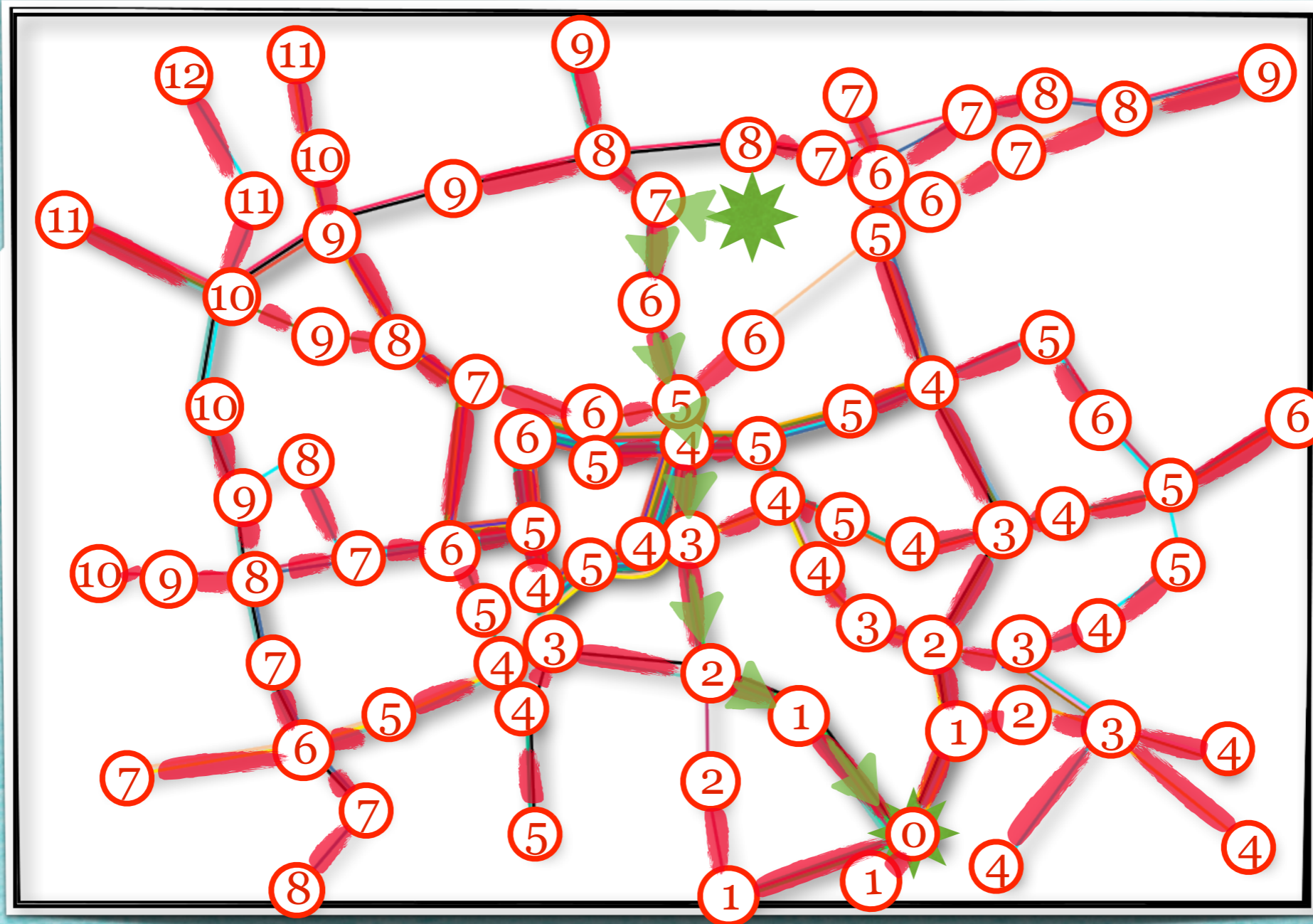
Breitensuche

Wellenreiten in Graphen



Breitensuche

Wellenreiten in Graphen



Breitensuche

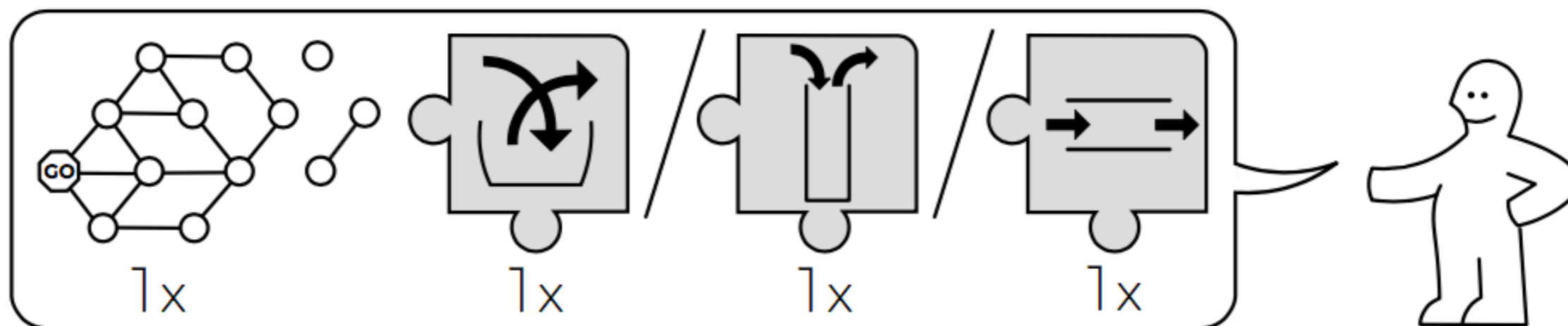
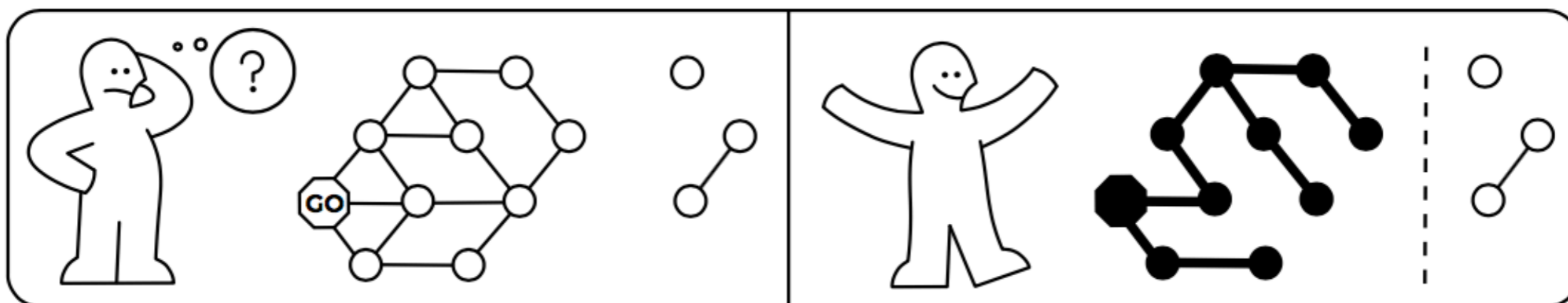
Mehr Details!

s.fekete@tu-bs.de

GRÅPH SCÄN

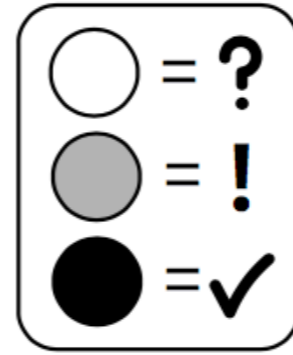
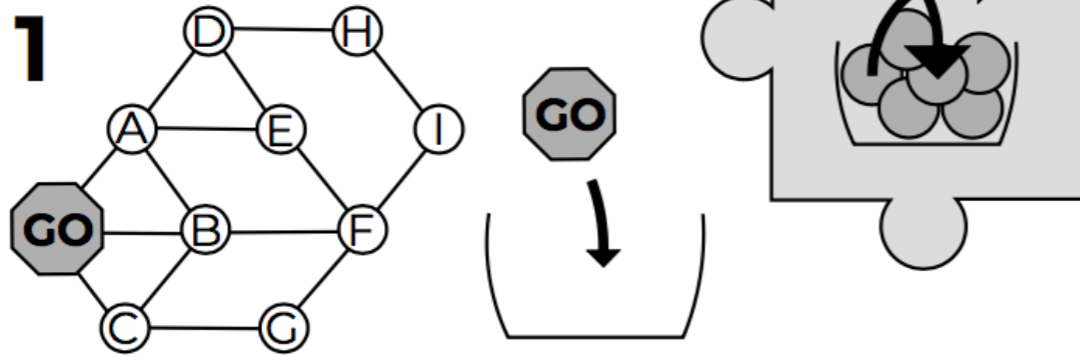
idea-instructions.com/graph-scan/
v1.0, CC by-nc-sa 4.0

IDEA

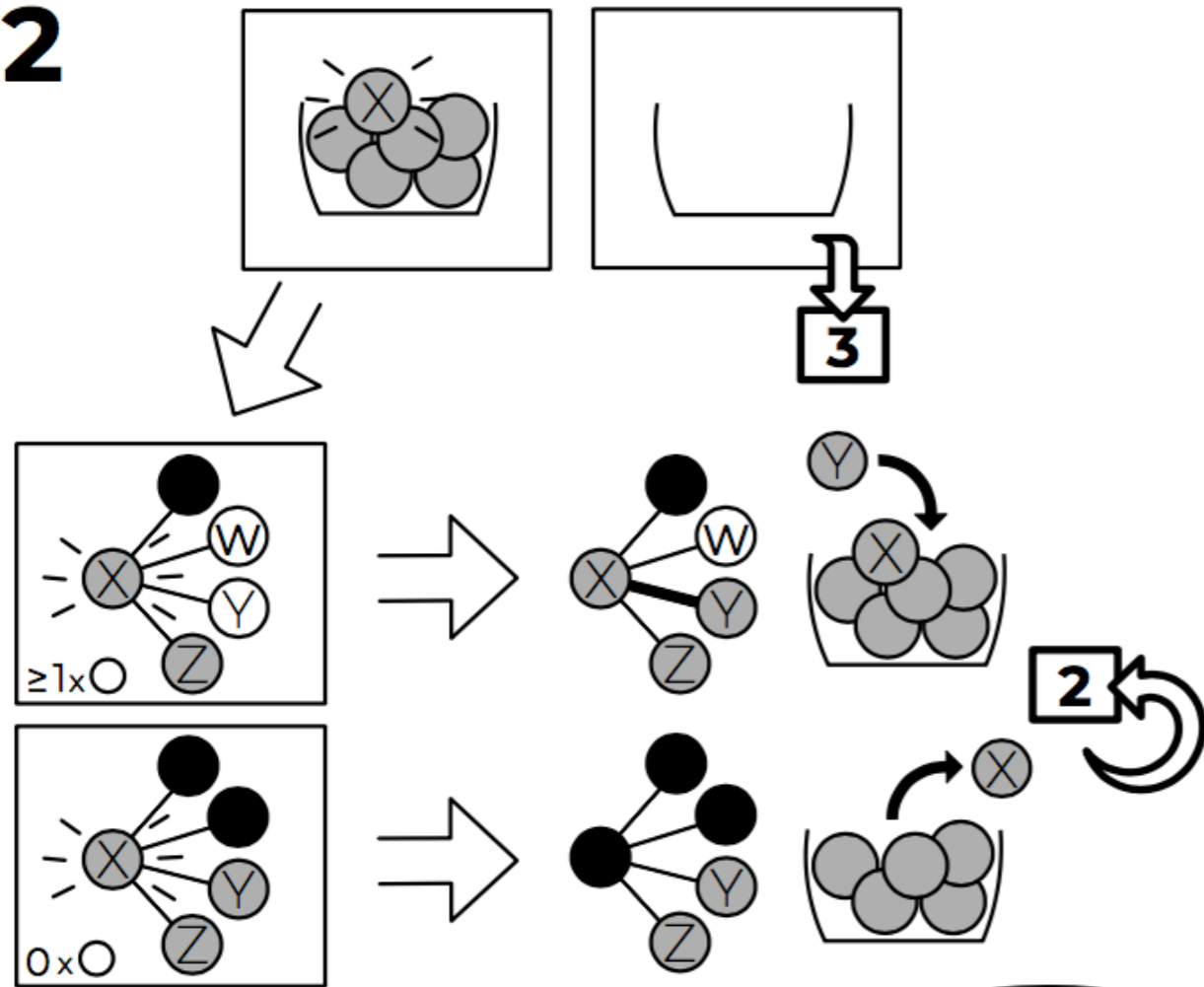


AD-HOC SEARCH

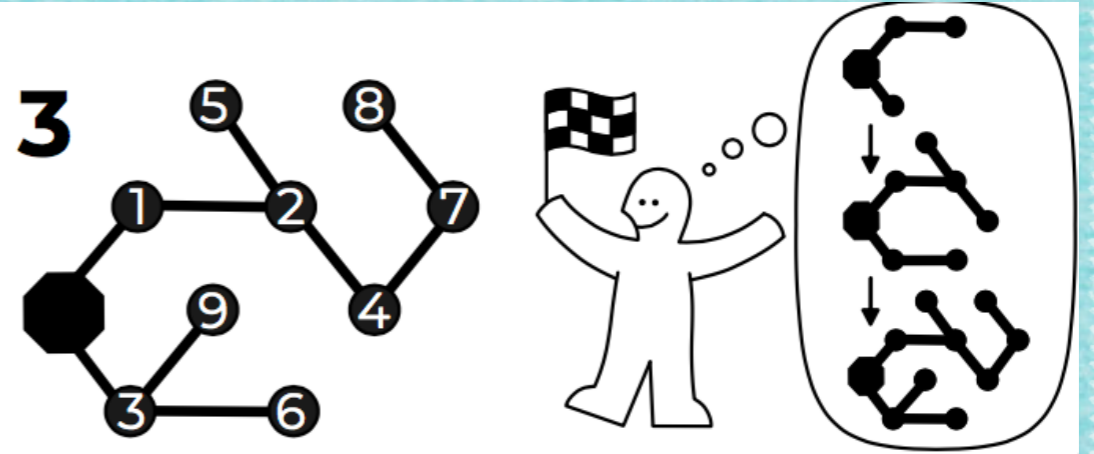
1



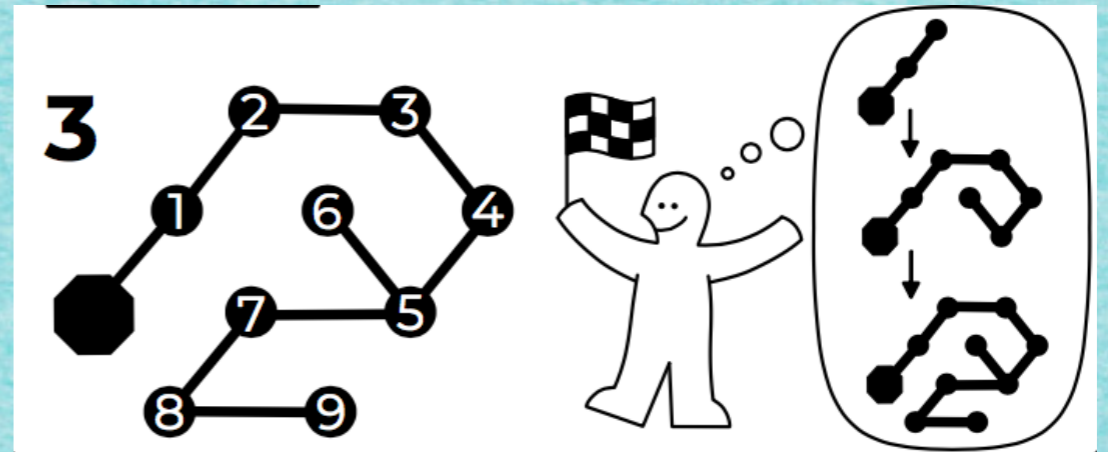
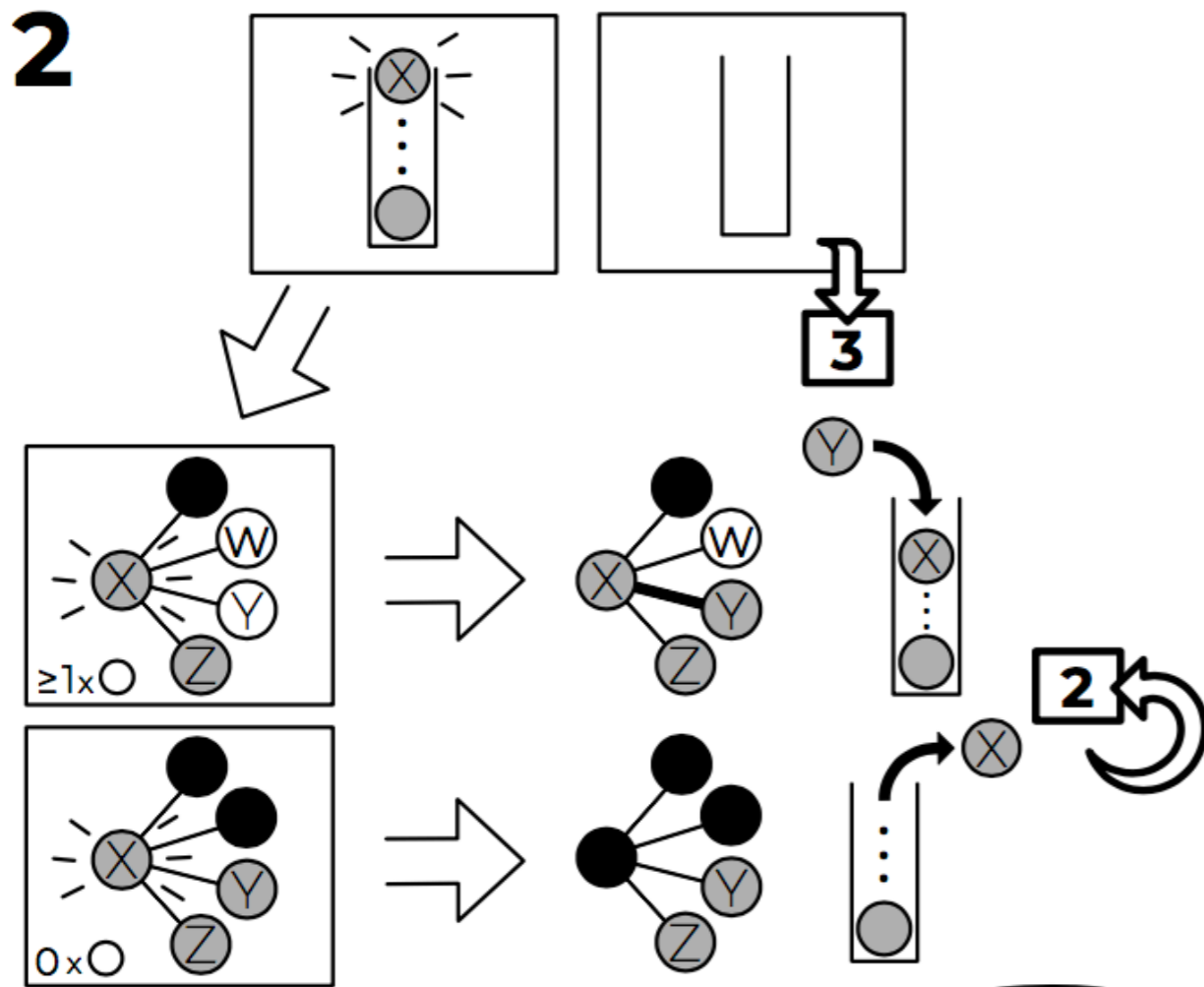
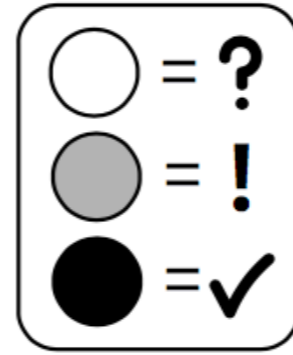
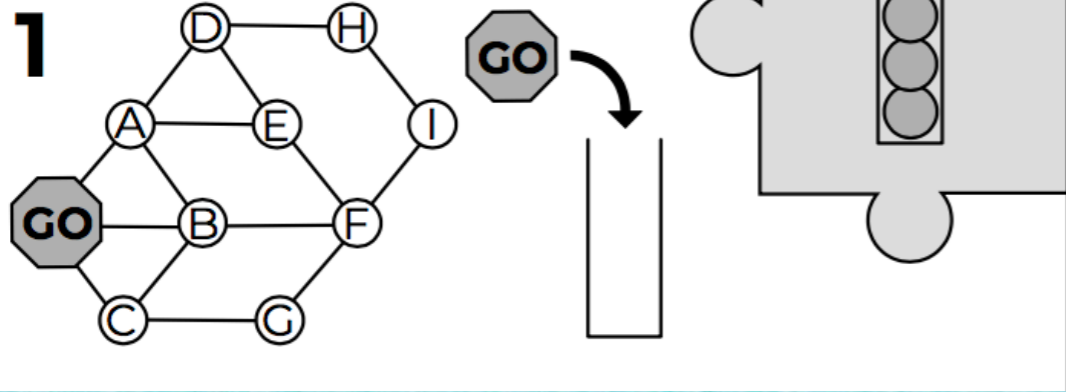
2



3

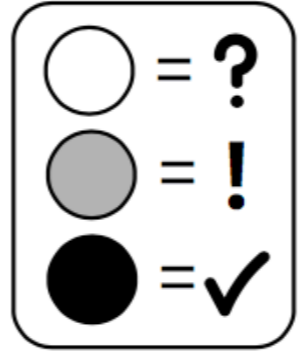
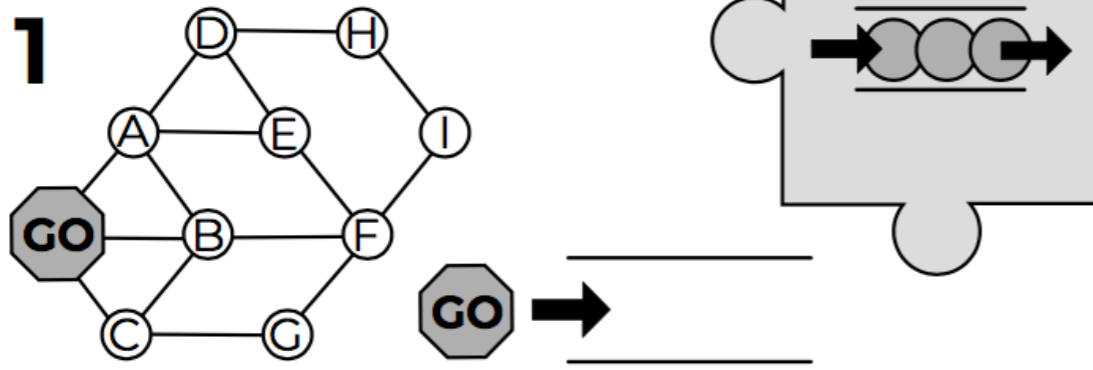


DEEP SEARCH

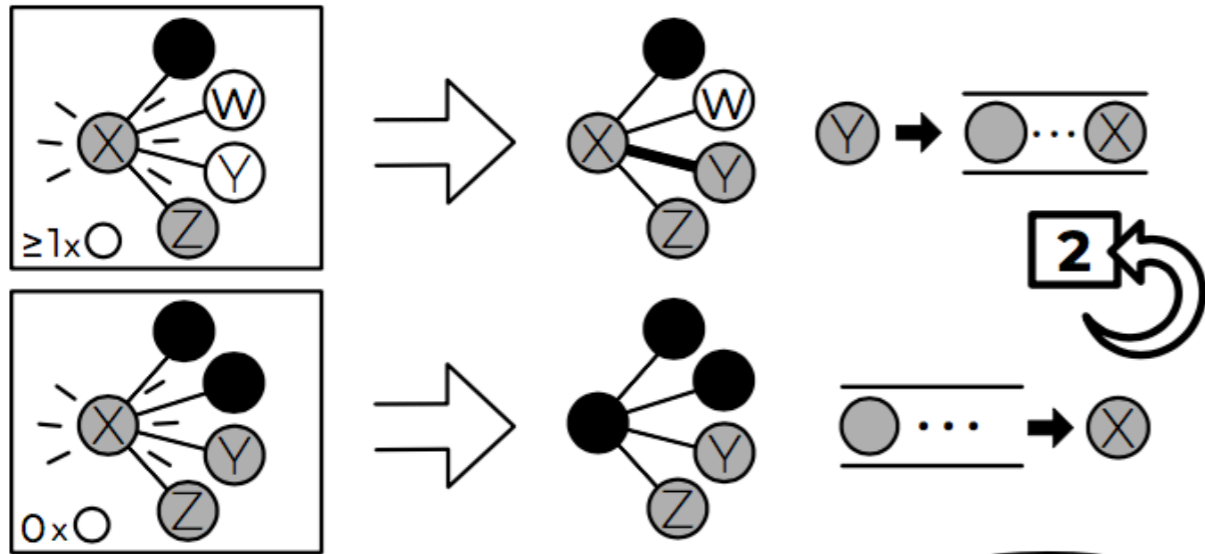
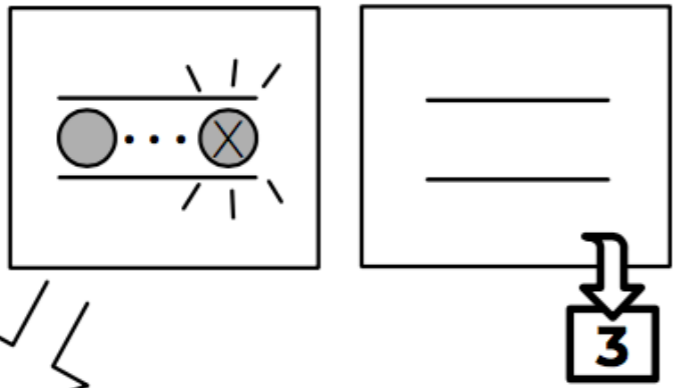


BROAD SEARCH

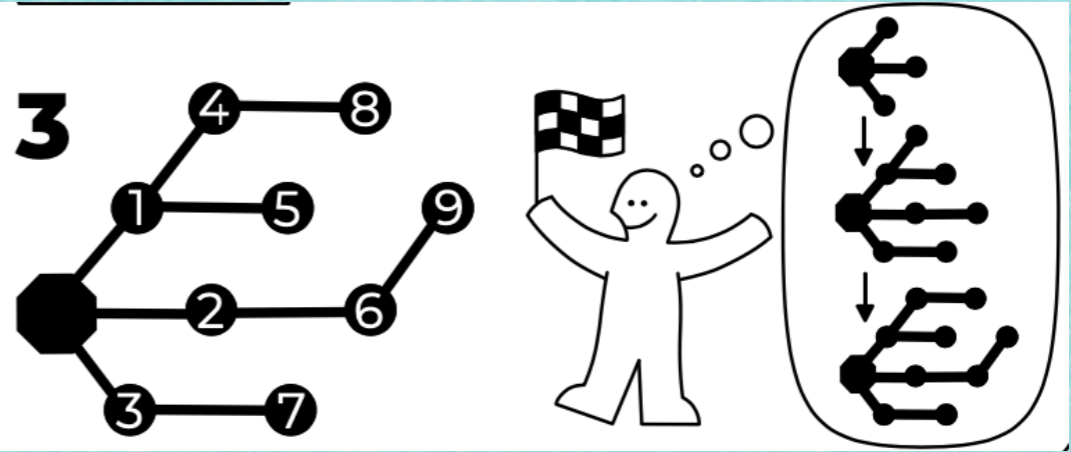
1



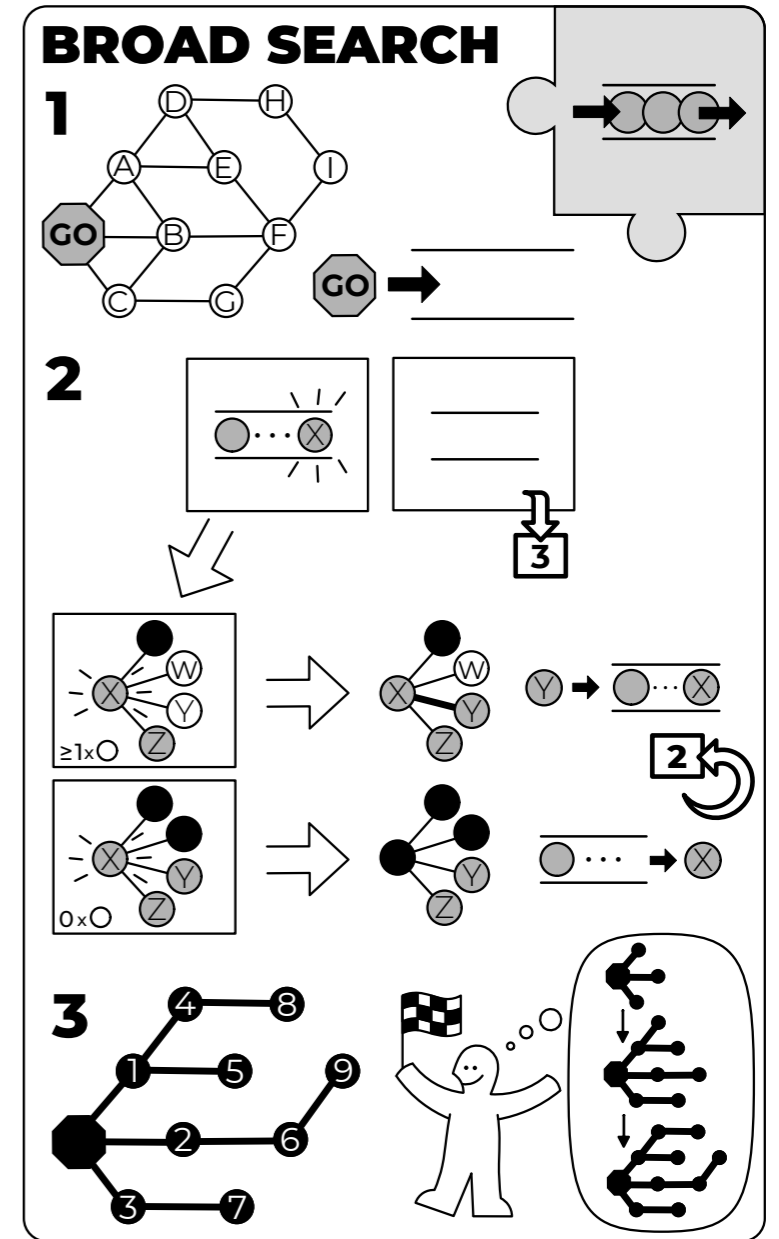
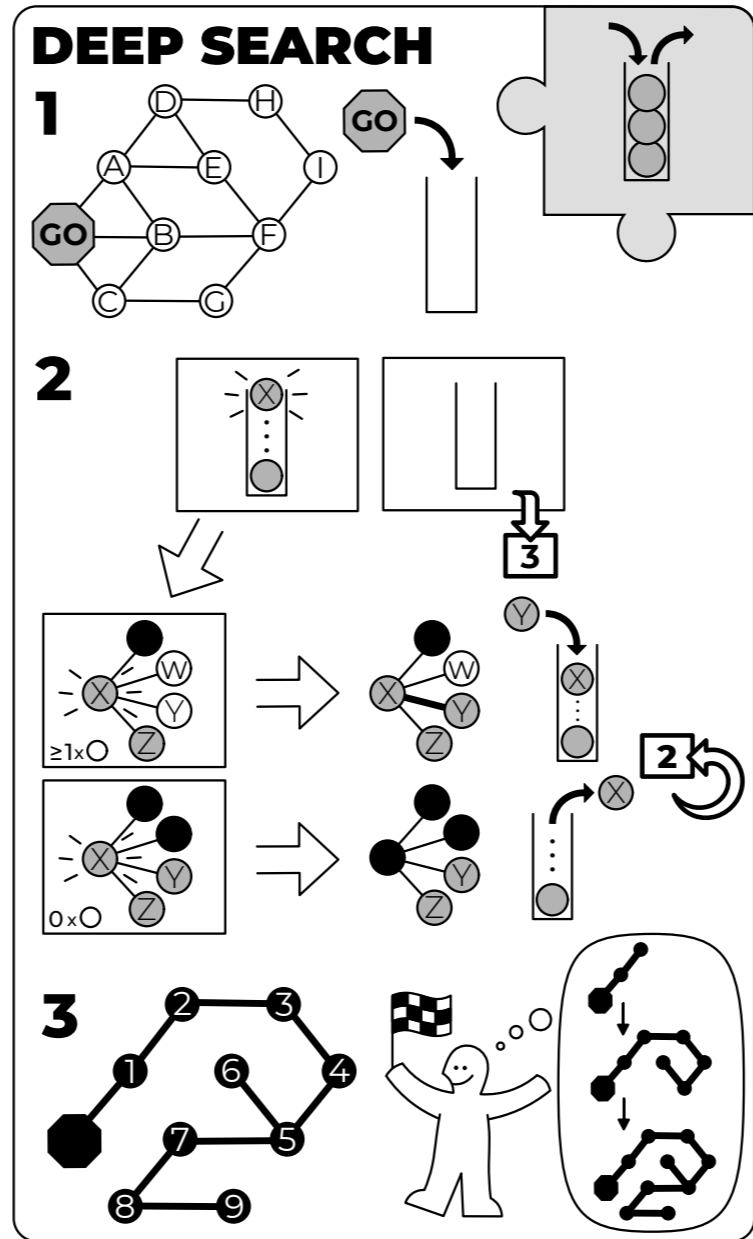
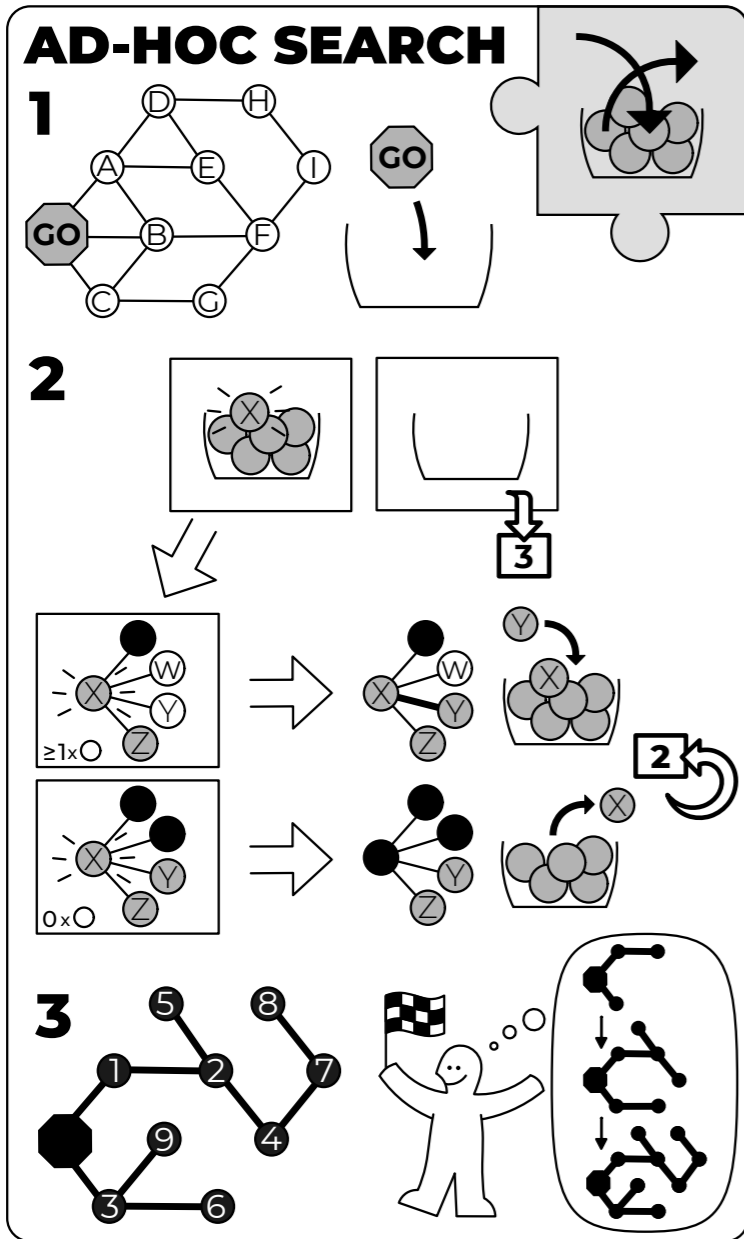
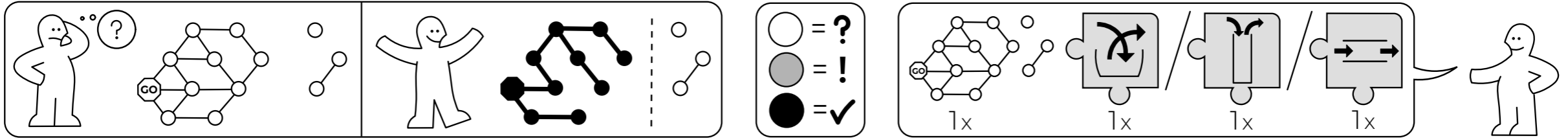
2



3

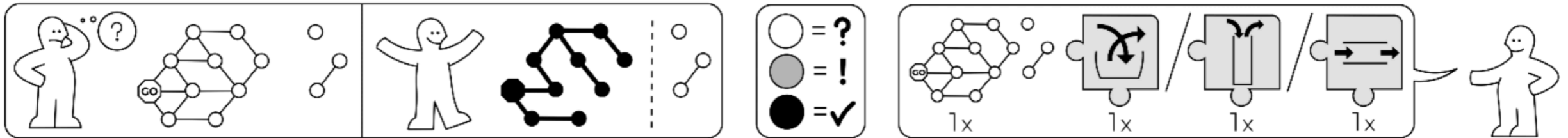


GRÅPH SKÄN



GRÅPH SKÄN

idea-instructions.com/graph-scan/
v1.3, CC by-nc-sa 4.0



AD-HOC SEARCH

1

2

3

DEEP SEARCH

1

2

3

BROAD SEARCH

1

2

3

Kapitelende!

s.fekete@tu-bs.de