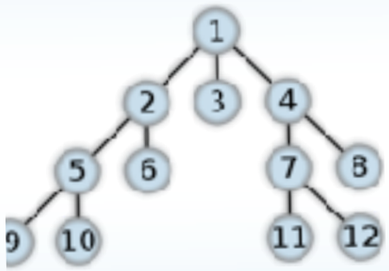


Kapitel 3.9: Eigenschaften von DFS und BFS

Algorithmen und Datenstrukturen
WS 2017/18

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:=∅
2. WHILE (R≠∅) 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\{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 u {w}, Y := Y u {w}, T := T u {e};

     }

}

Auf die Schnelle mit der Welle

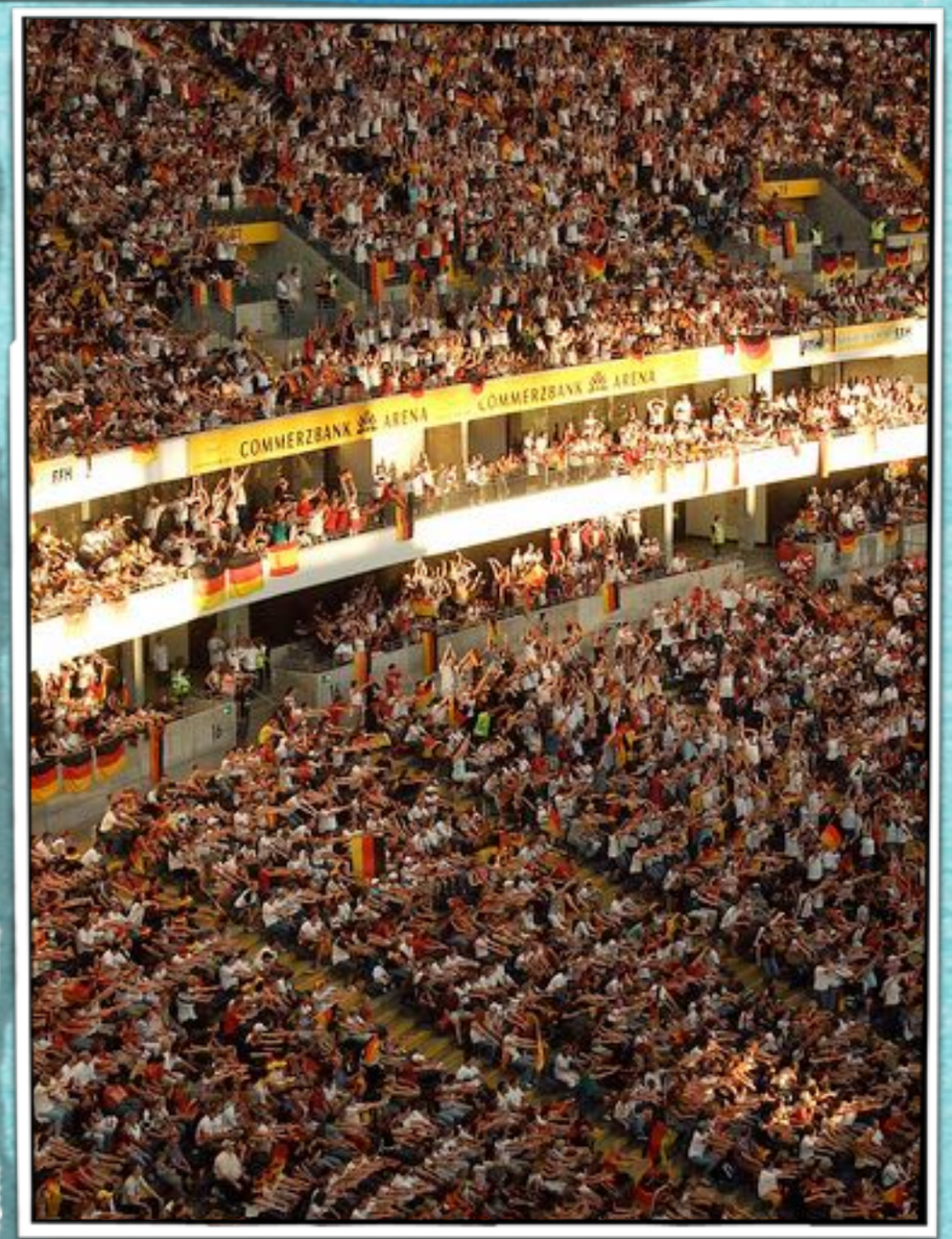
A. LOS bei „NULL“

B. Bis „ANGEKOMMEN!“:

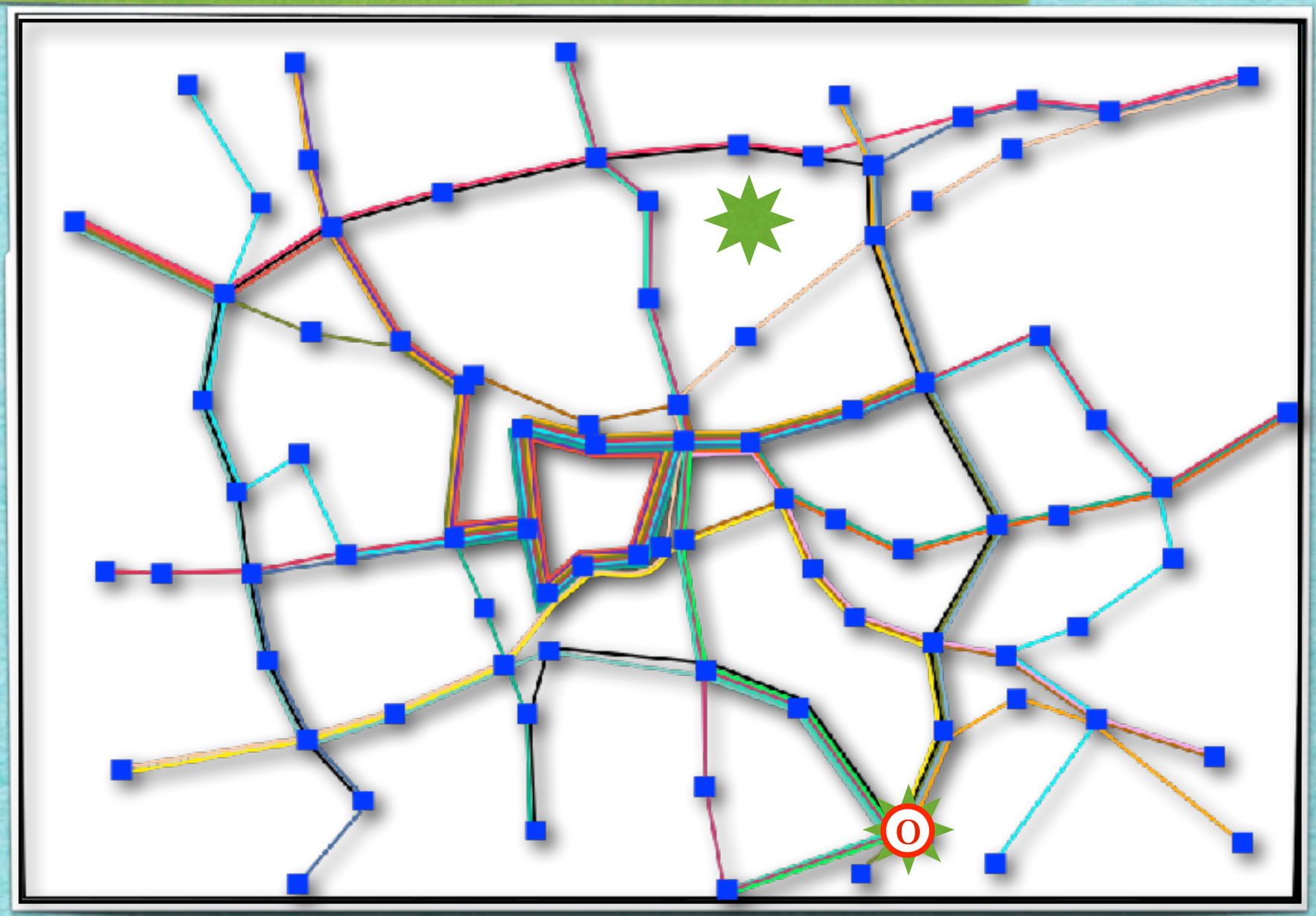
- Solange du noch nicht aufgestanden warst:
  - Wenn ein oder mehrere direkte Nachbarn aufstehen:
    1. Einen dieser Nachbarn merken
    2. In der nächsten Runde:
       2.1. aufstehen
       2.2. Zahl merken
    3. In der übernächsten Runde hinsetzen

C. Nach „ANGEKOMMEN!“:

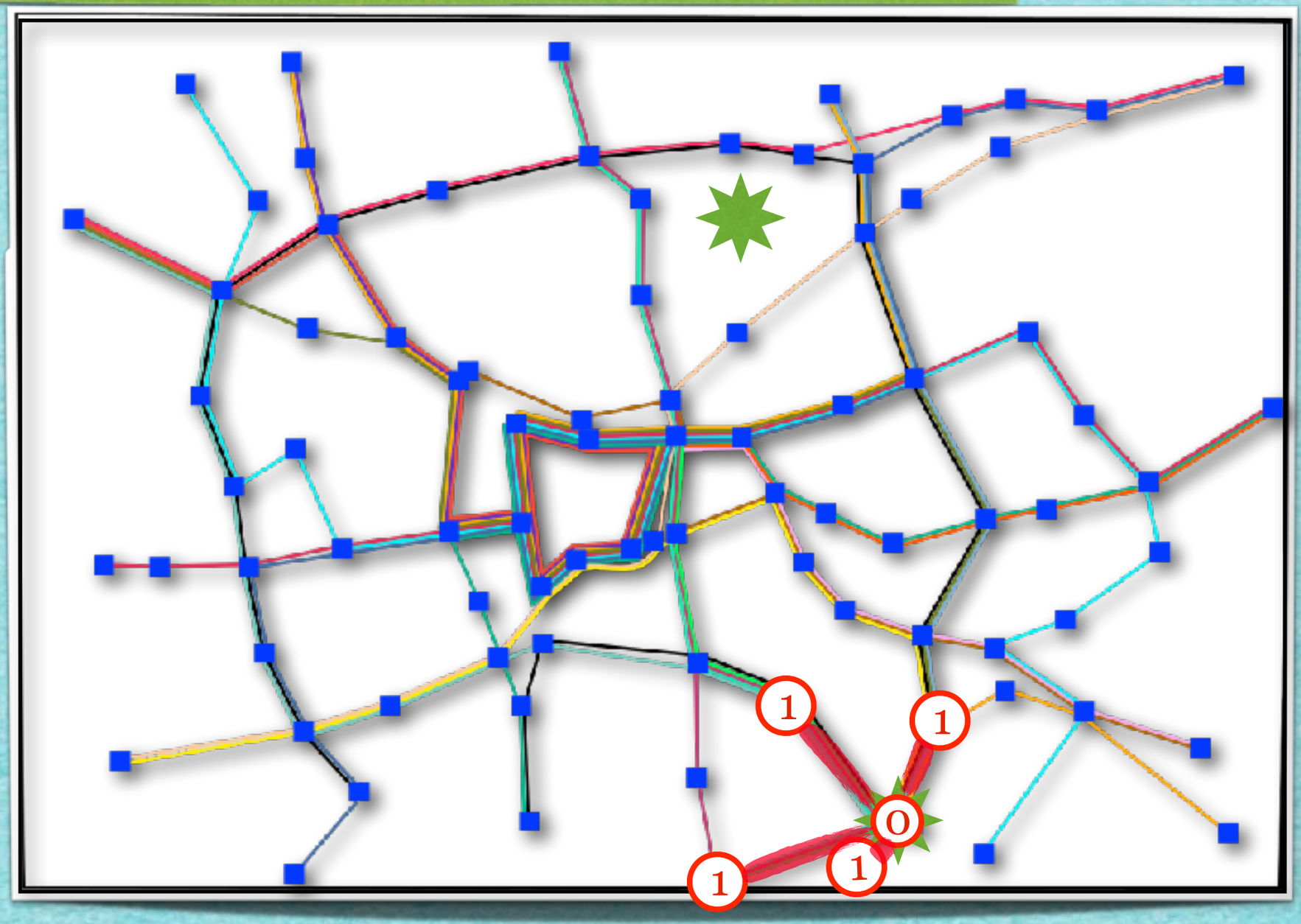
- Auf gemerkten Nachbarn zeigen



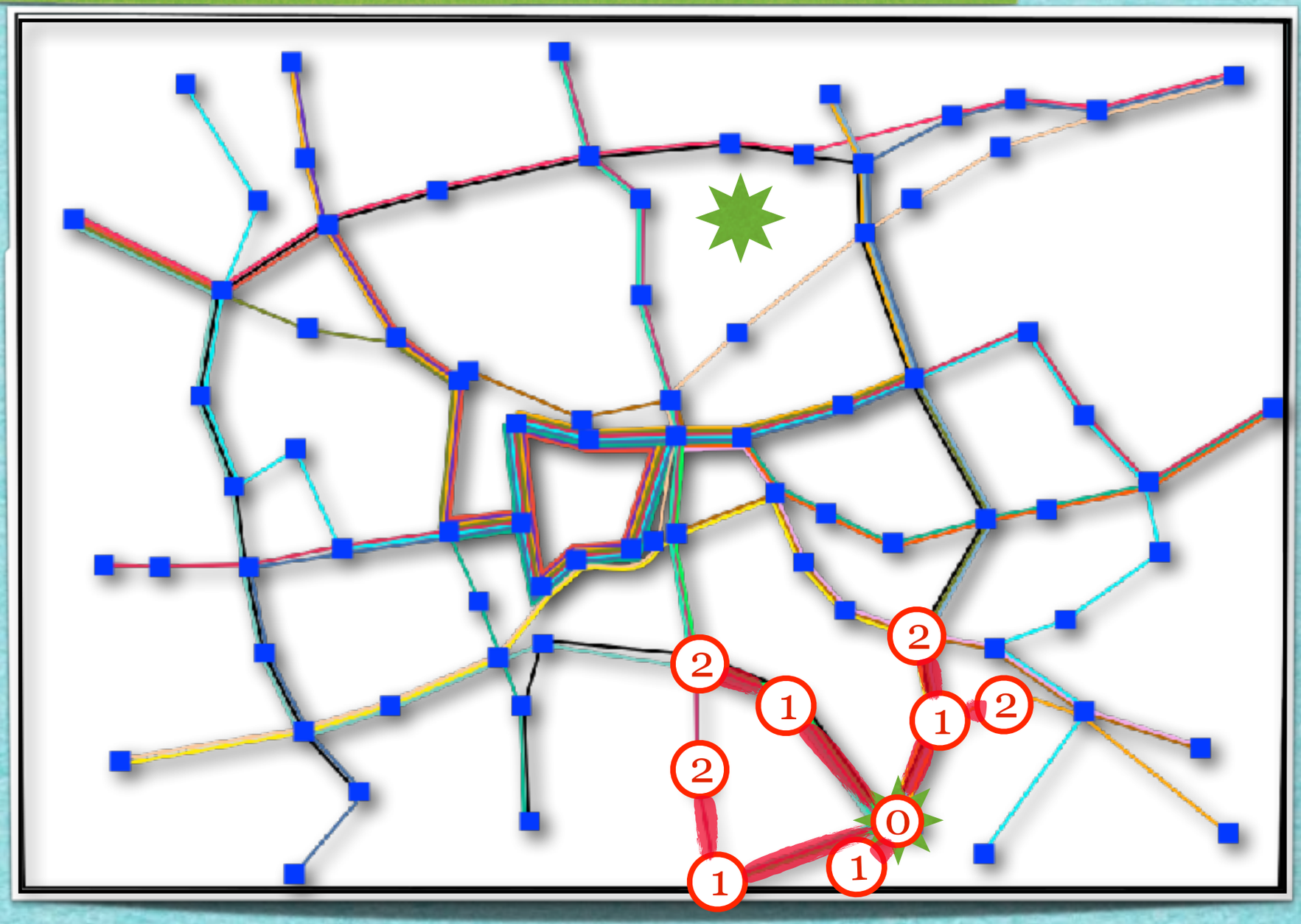
Wellenreiten in Graphen



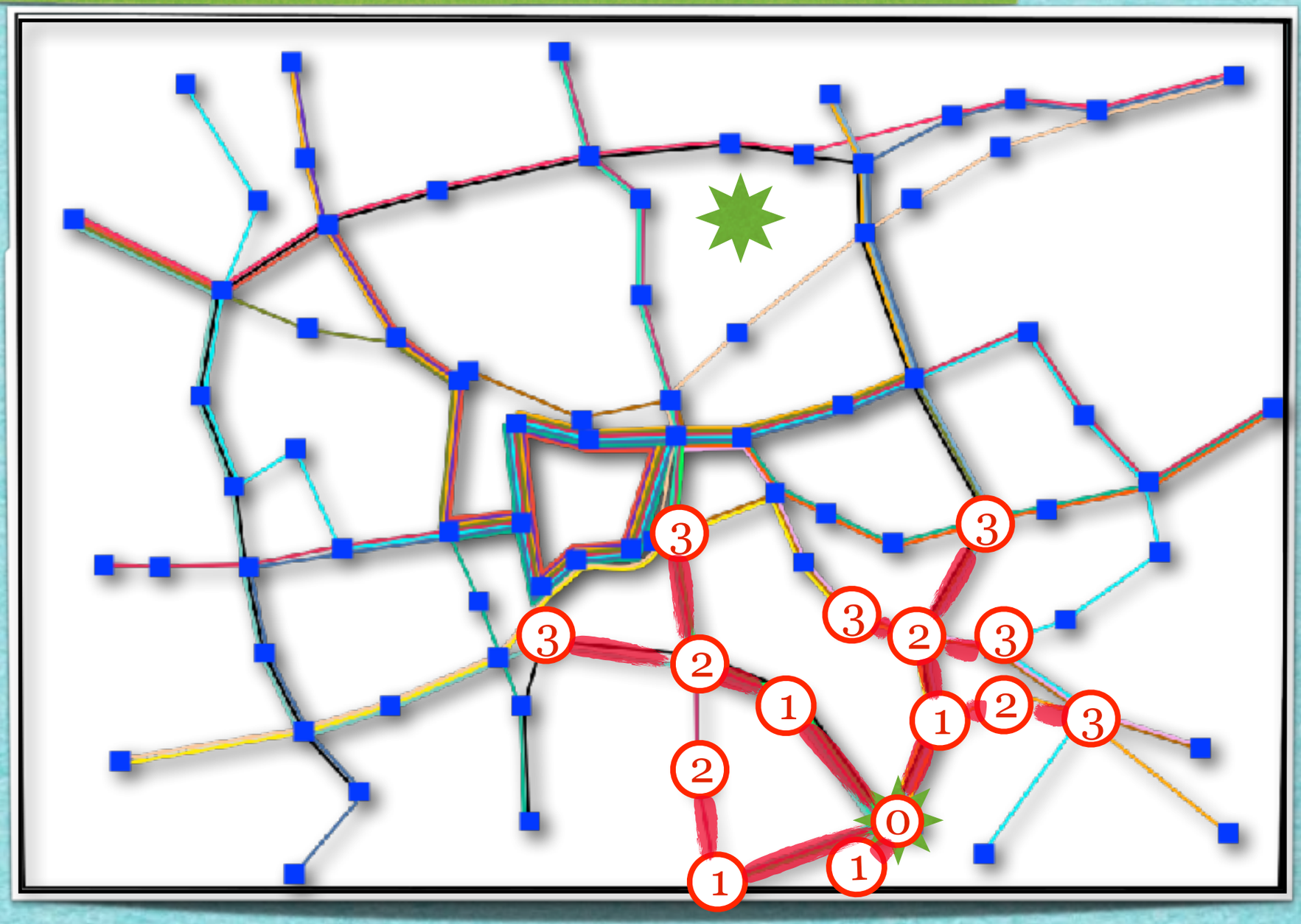
Wellenreiten in Graphen



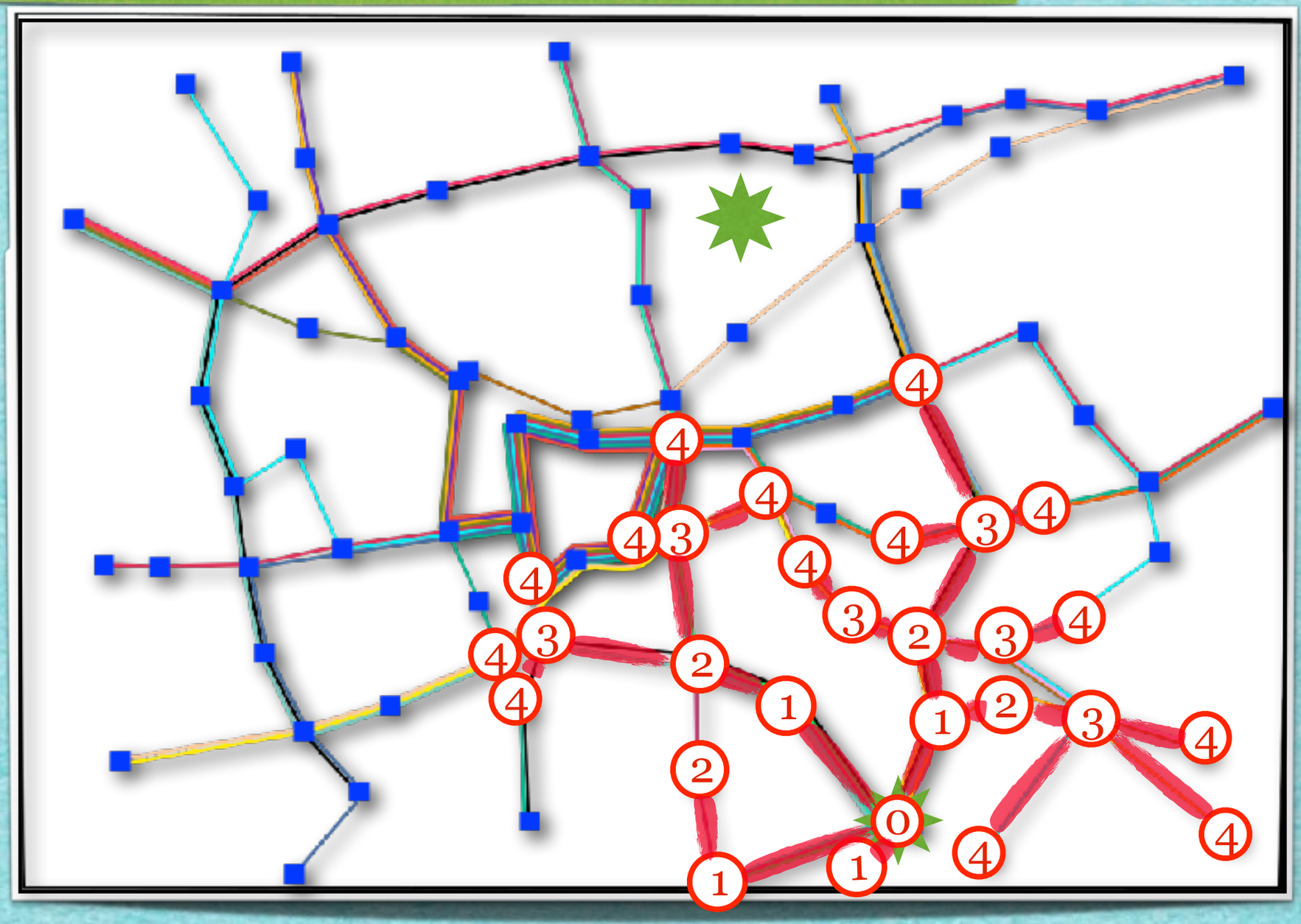
Wellenreiten in Graphen



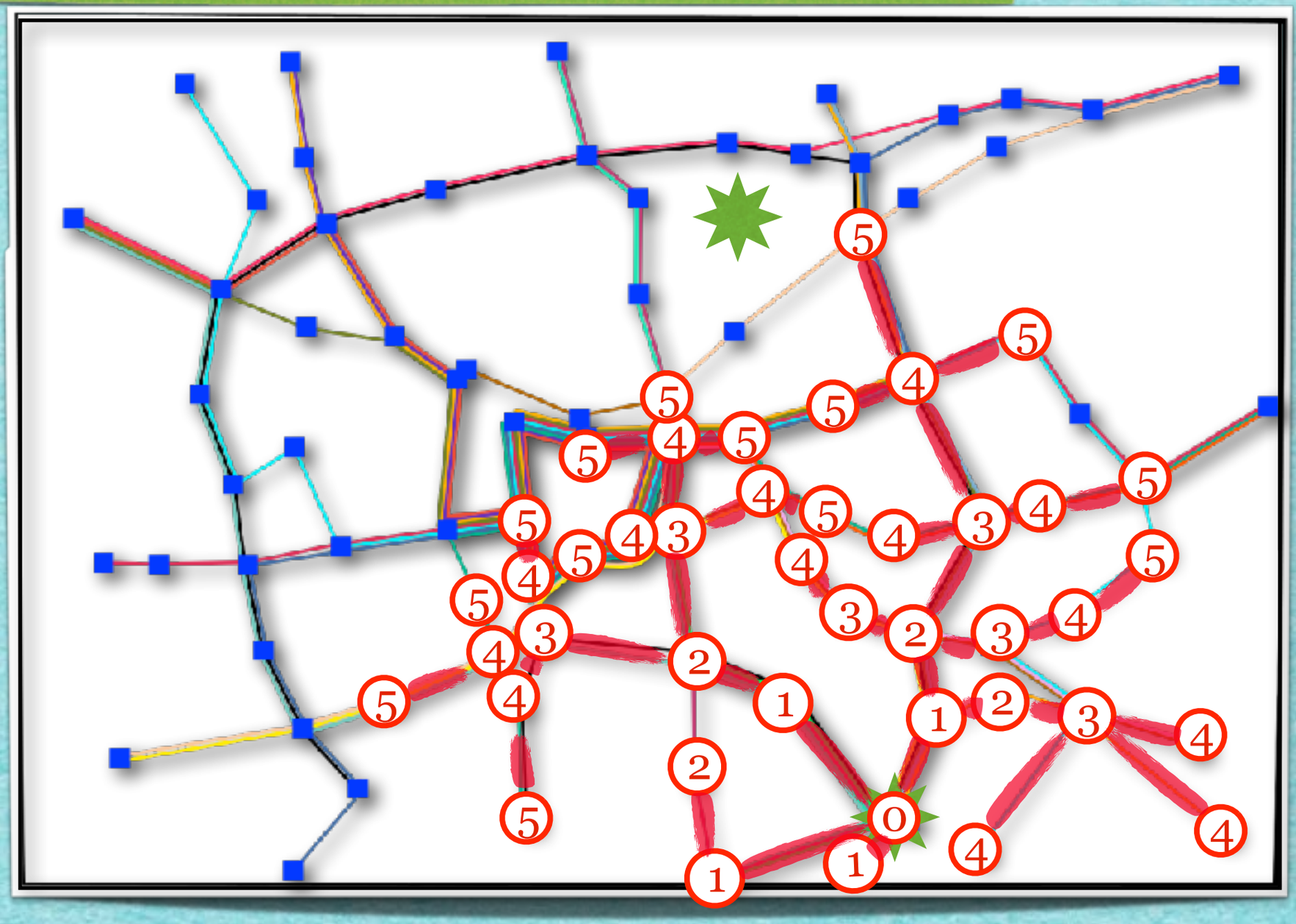
Wellenreiten in Graphen



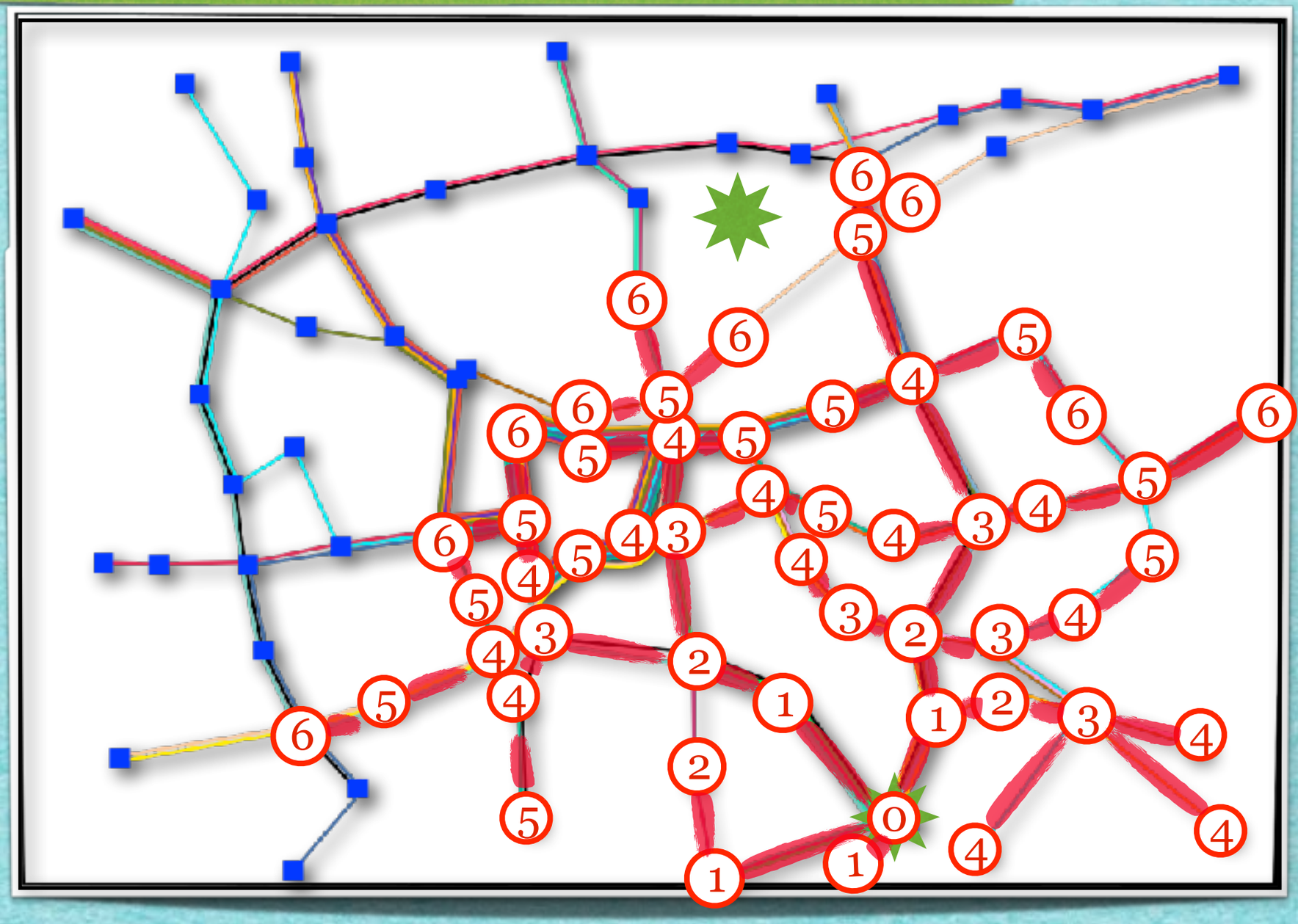
Wellenreiten in Graphen



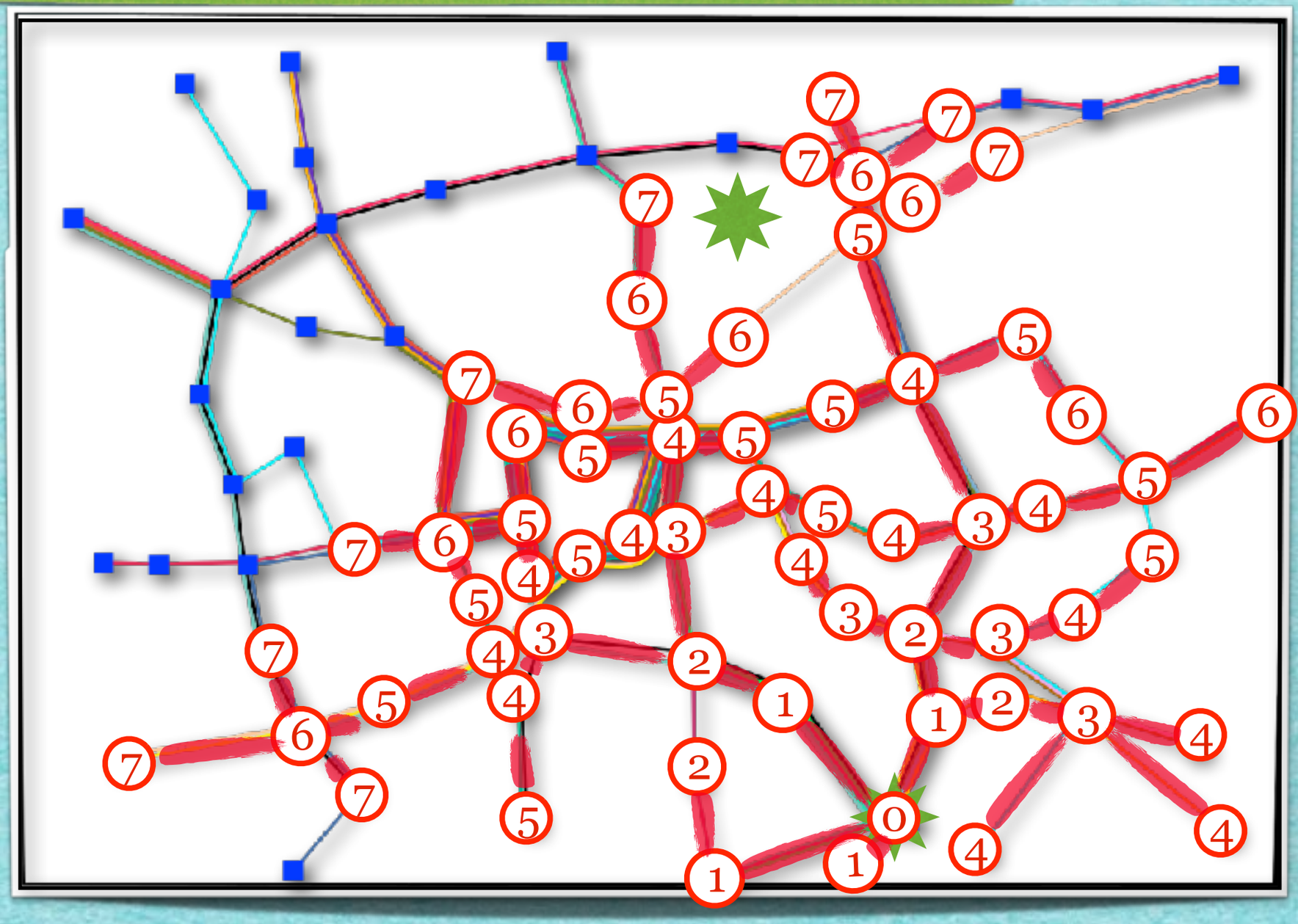
Wellenreiten in Graphen



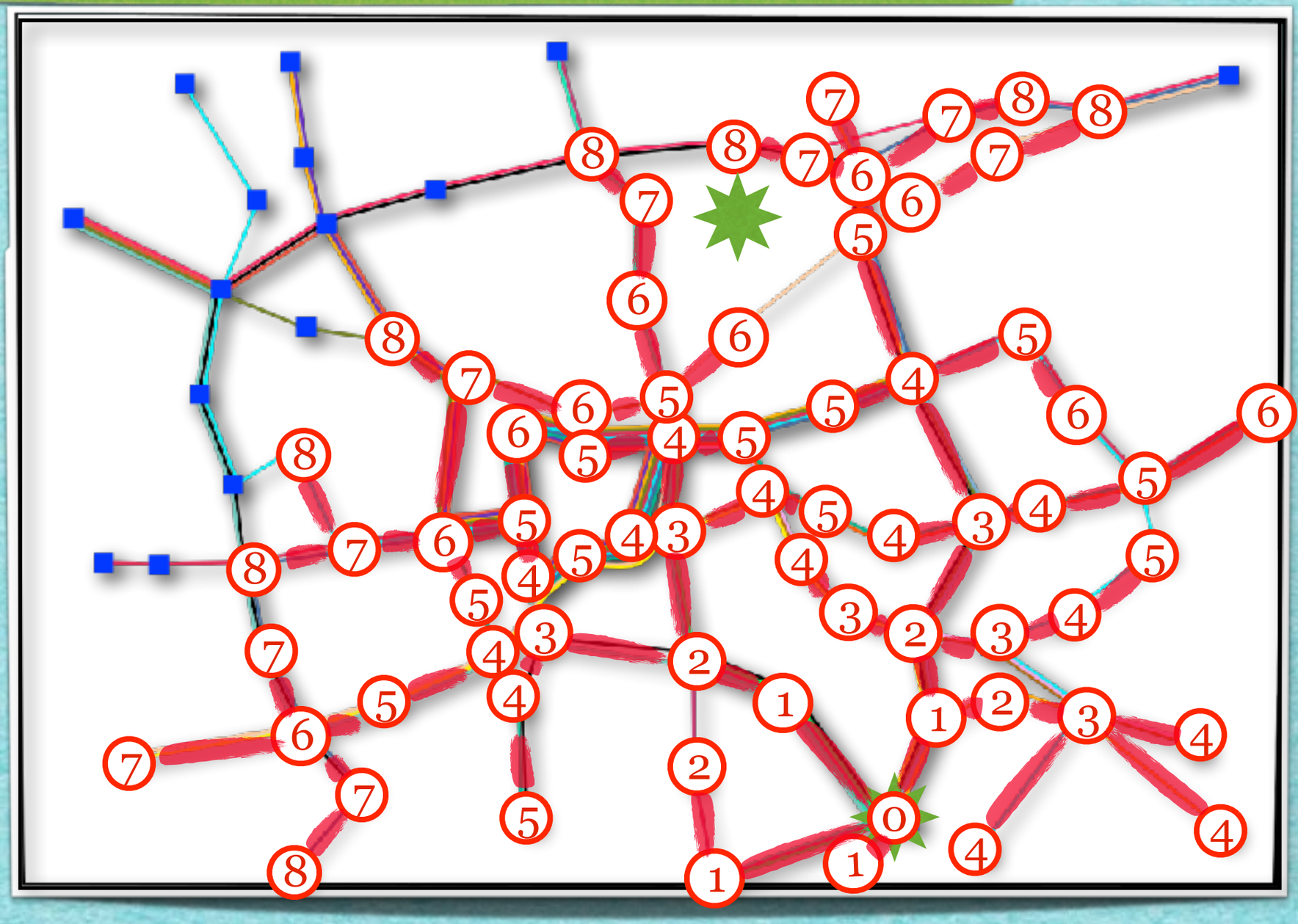
Wellenreiten in Graphen



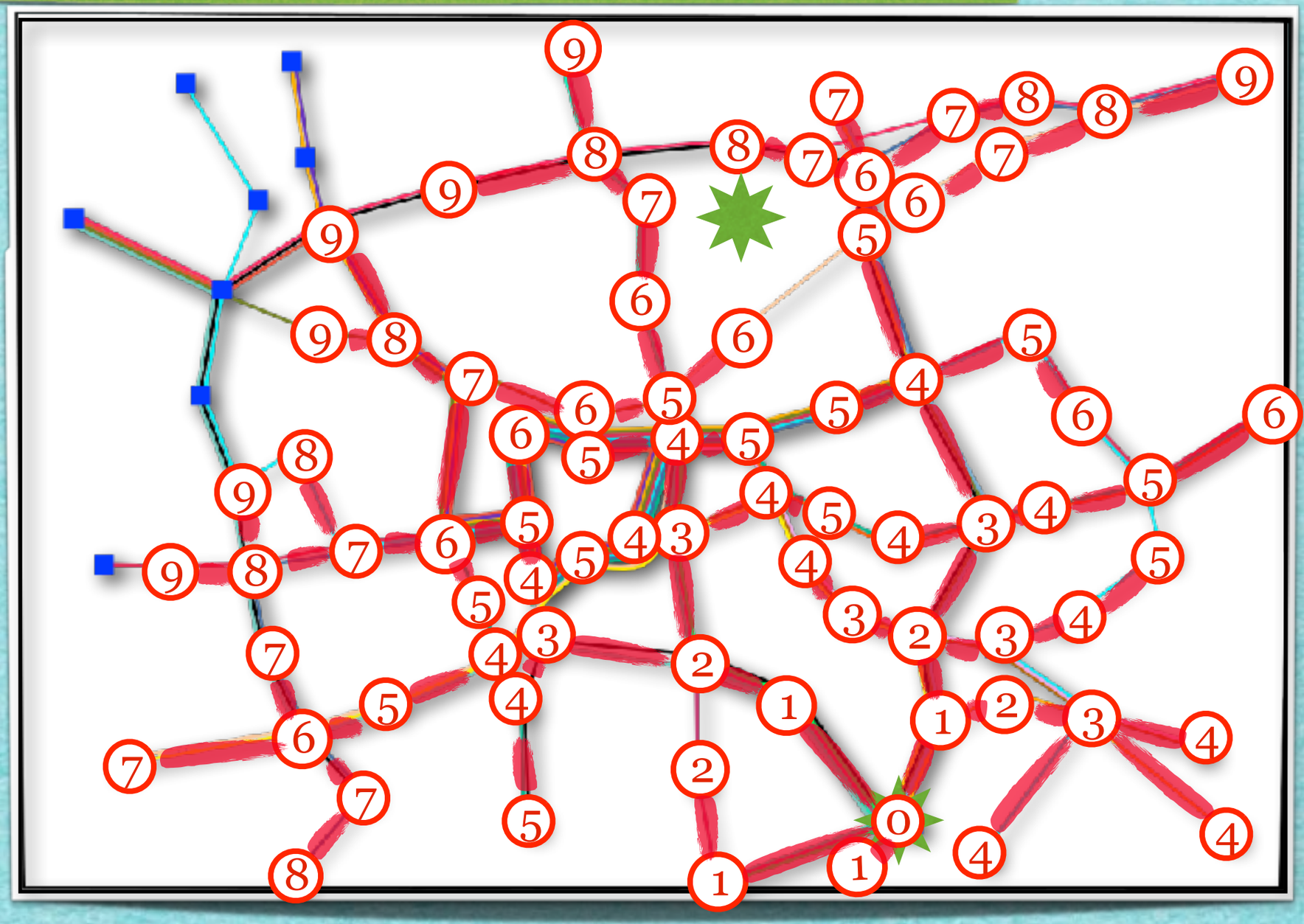
Wellenreiten in Graphen



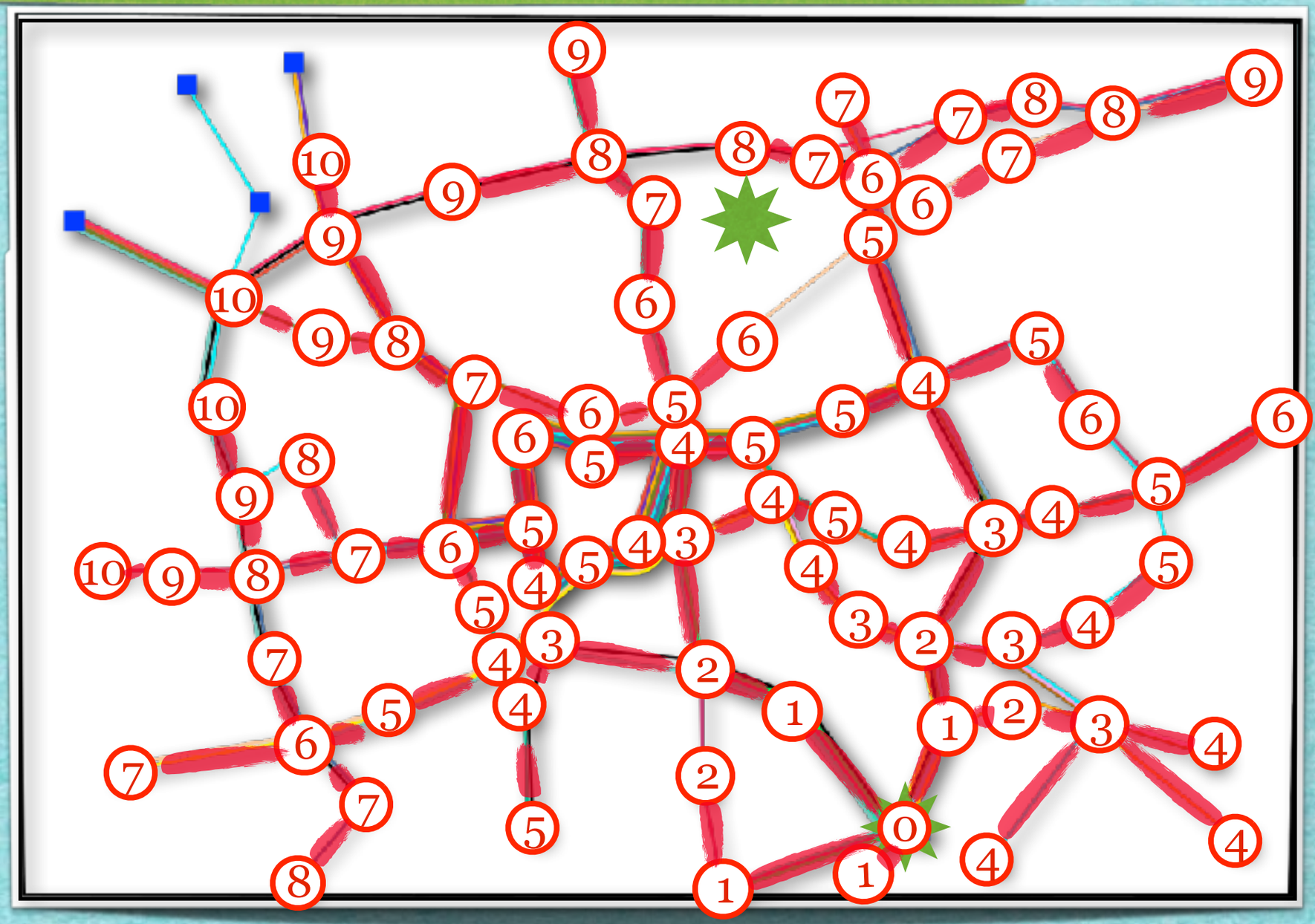
Wellenreiten in Graphen



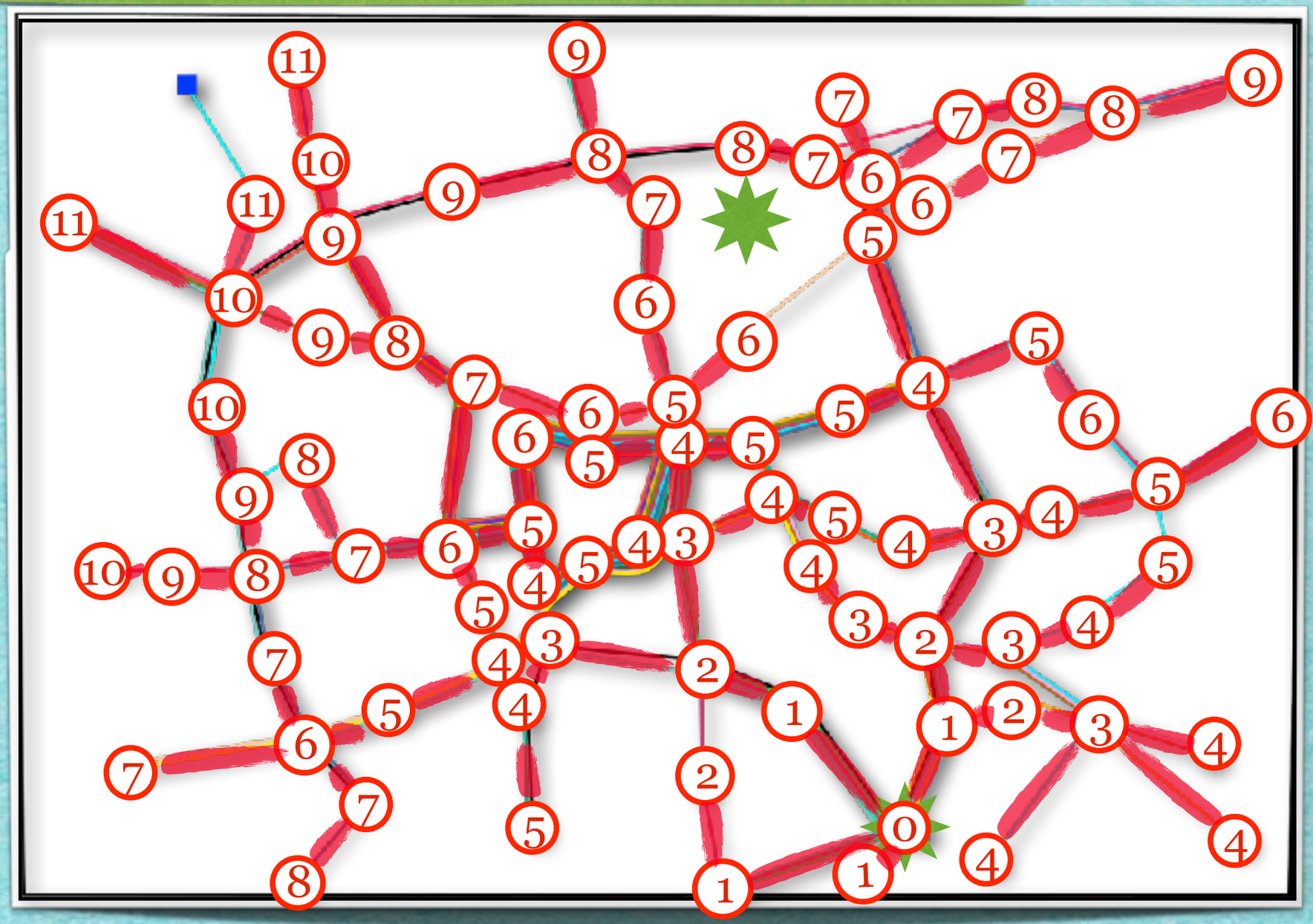
Wellenreiten in Graphen



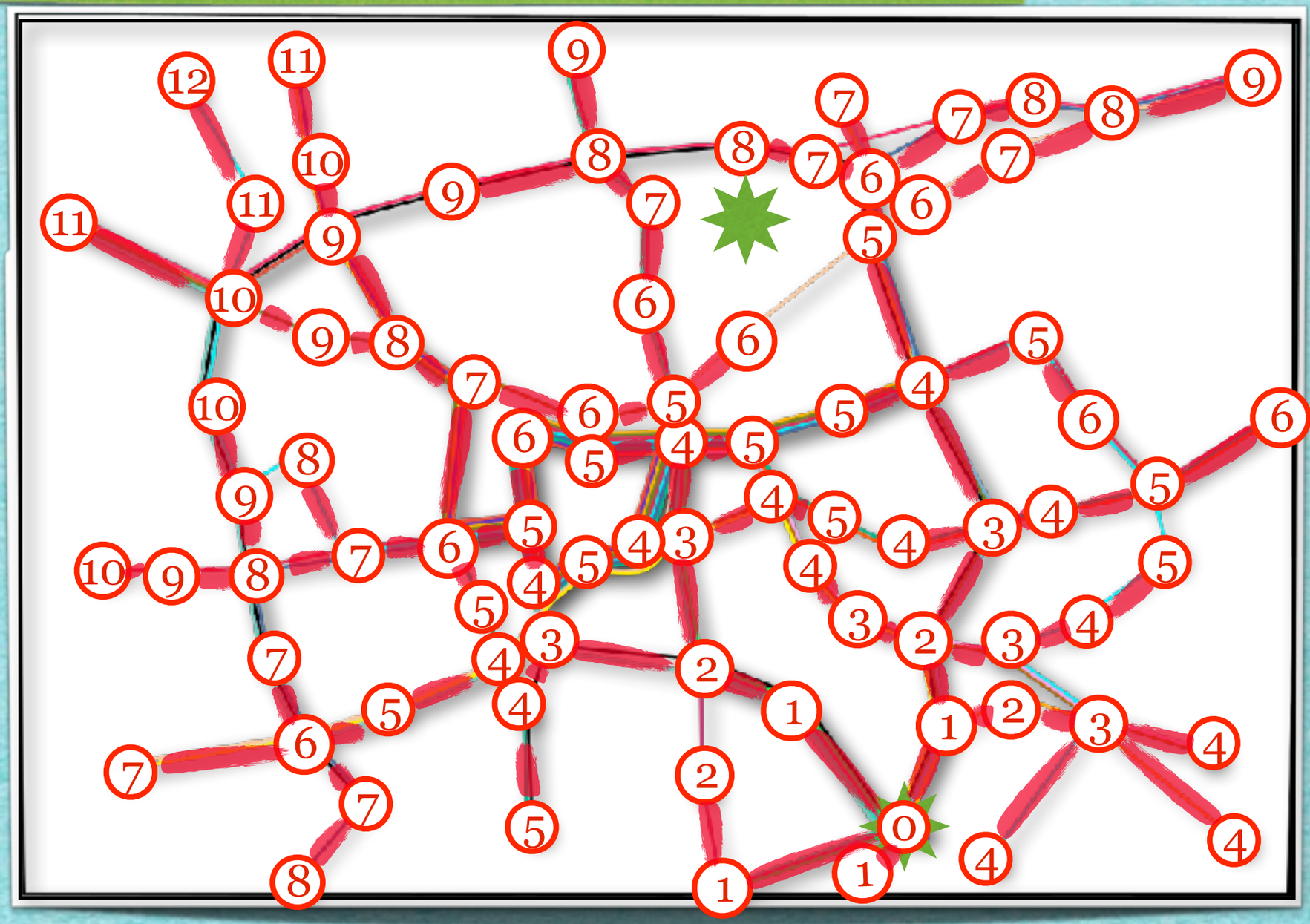
Wellenreiten in Graphen



Wellenreiten in Graphen

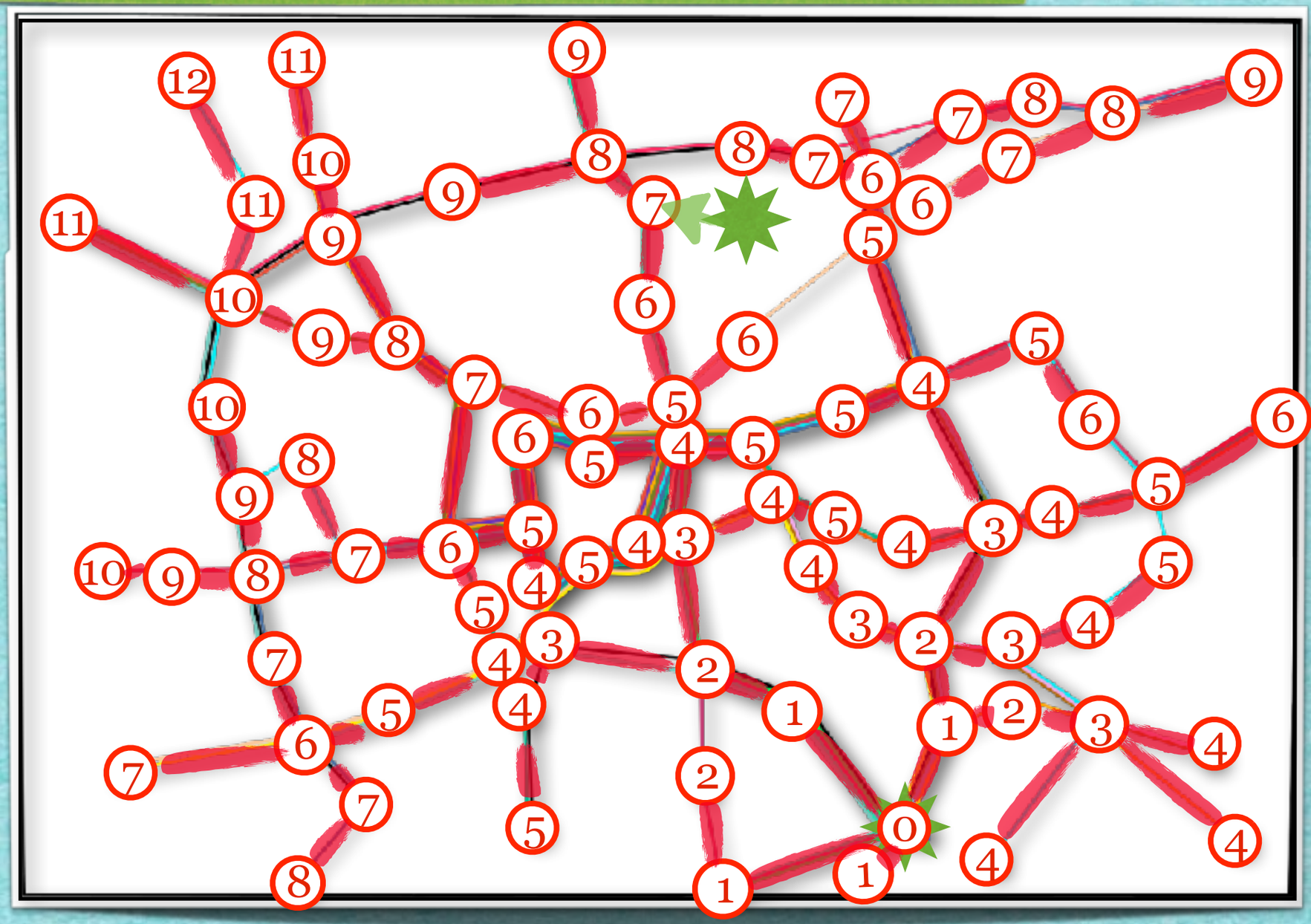


Wellenreiten in Graphen



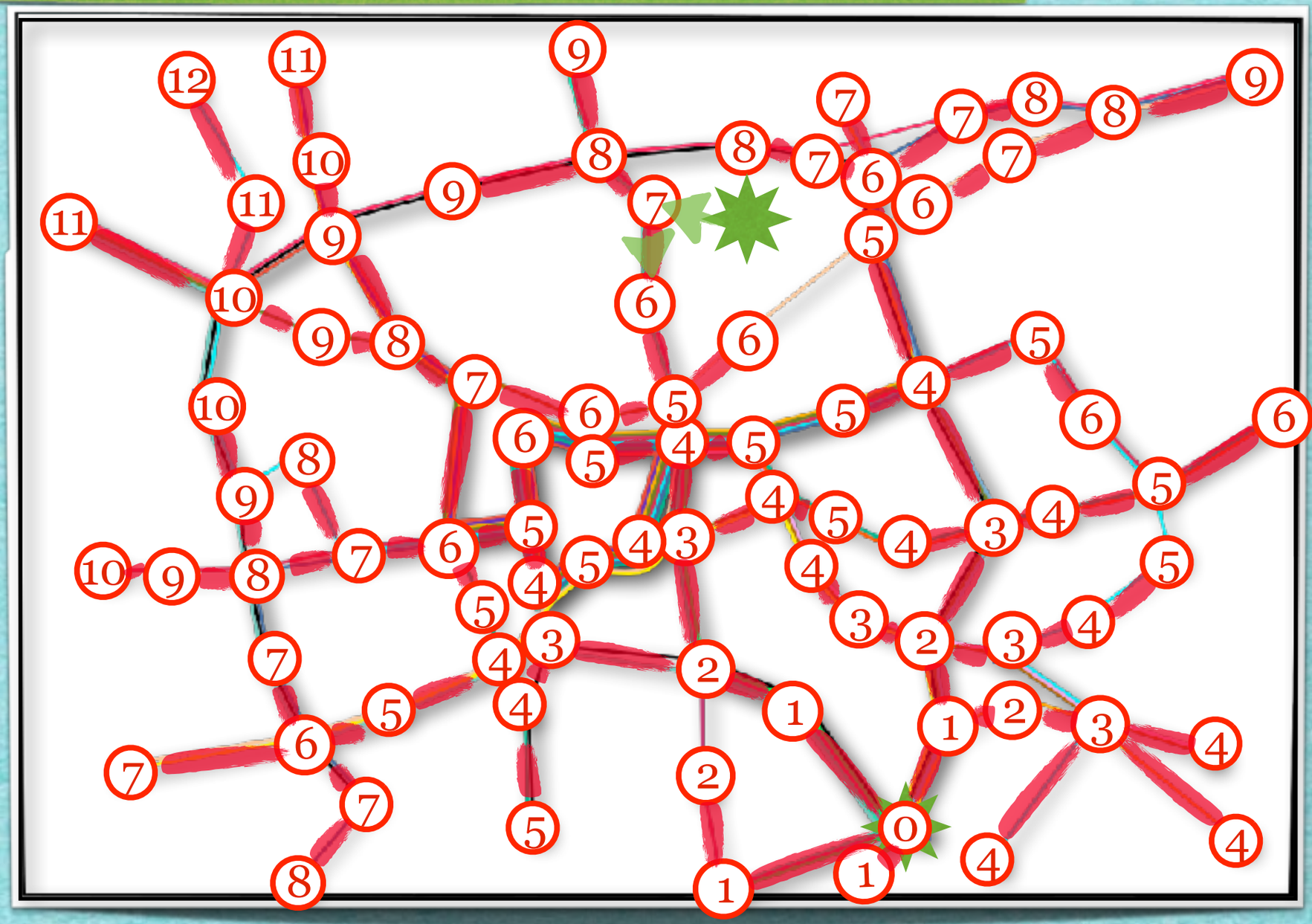
Breitensuche

Wellenreiten in Graphen



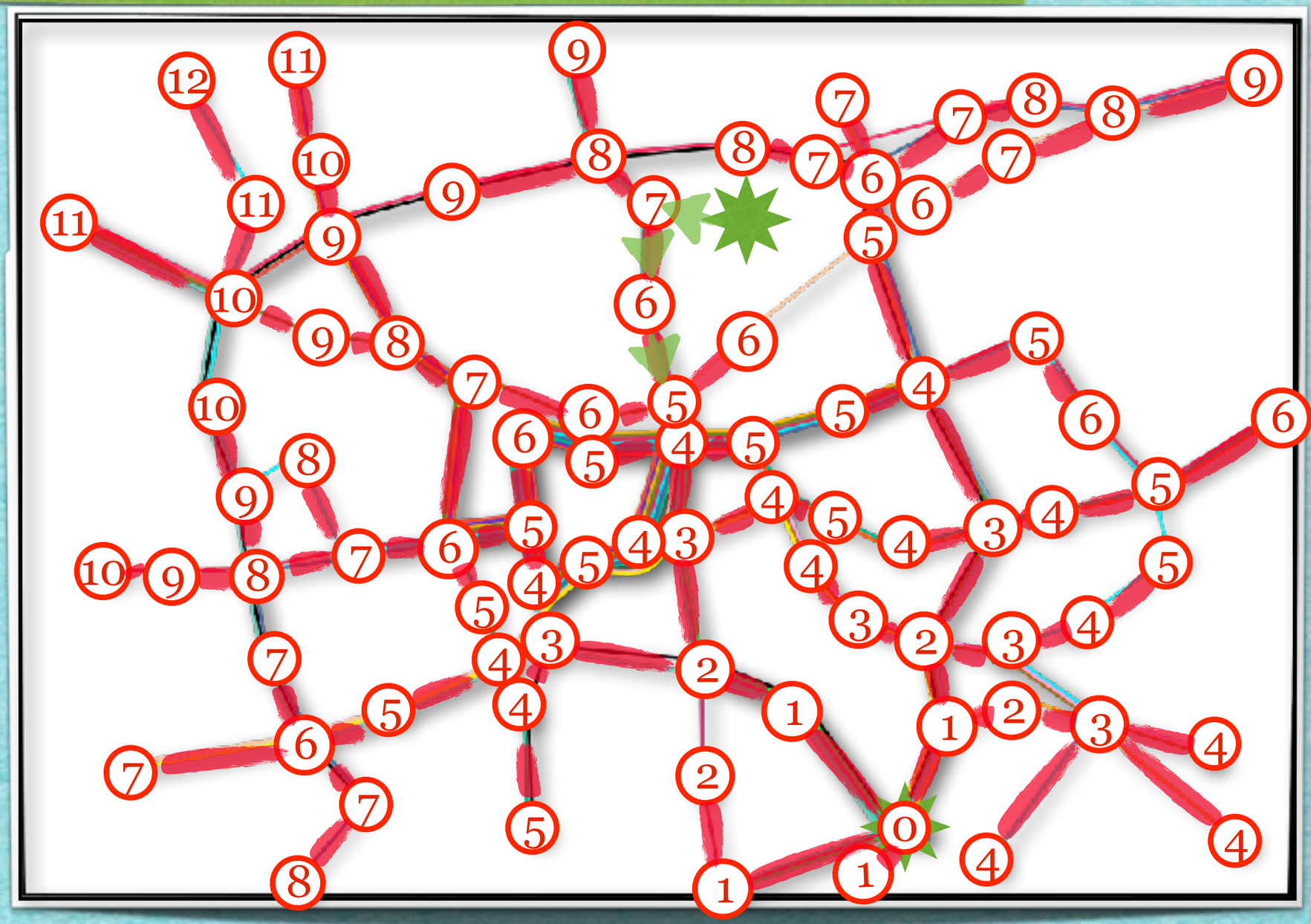
Breitensuche

Wellenreiten in Graphen



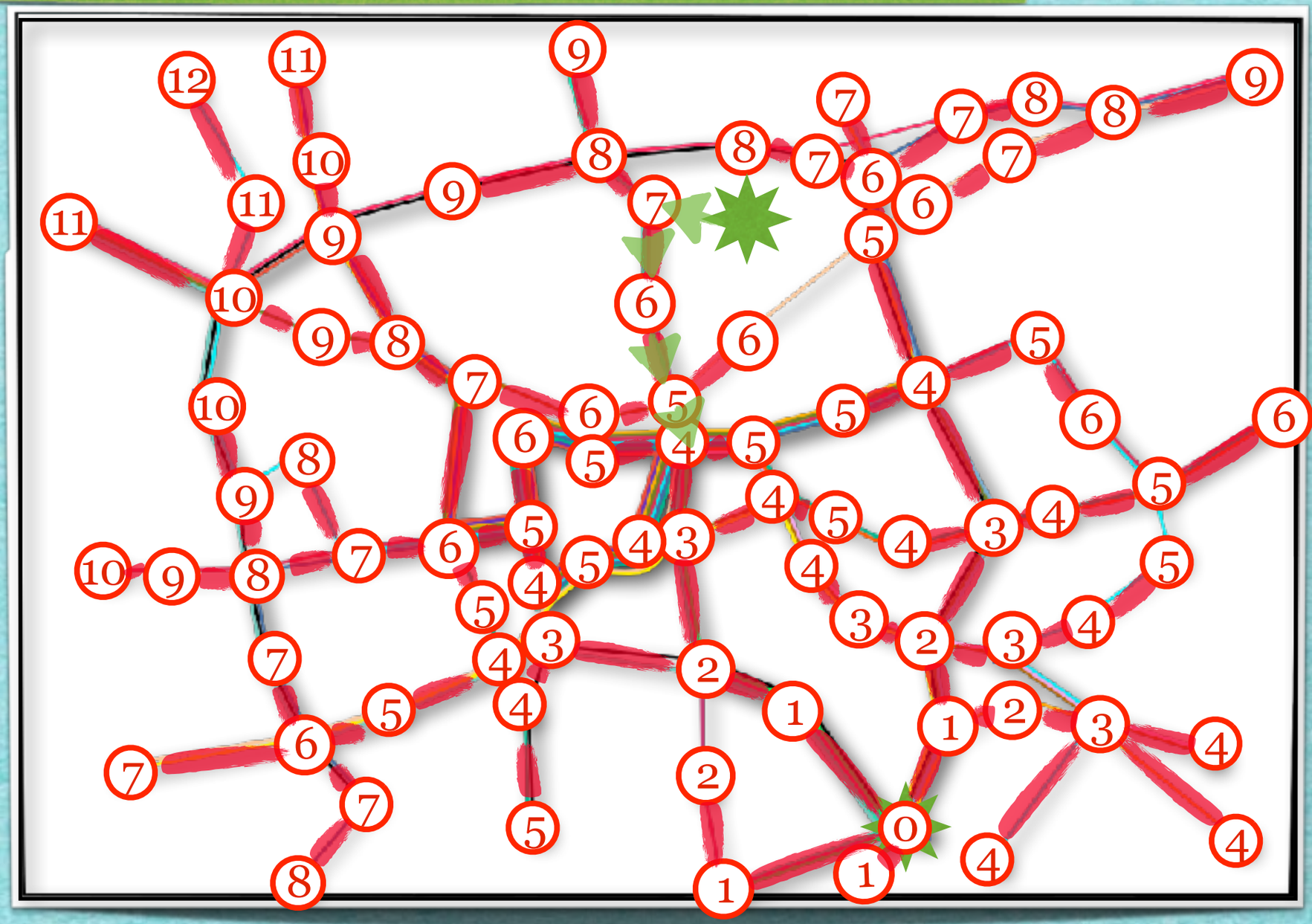
Breitensuche

Wellenreiten in Graphen



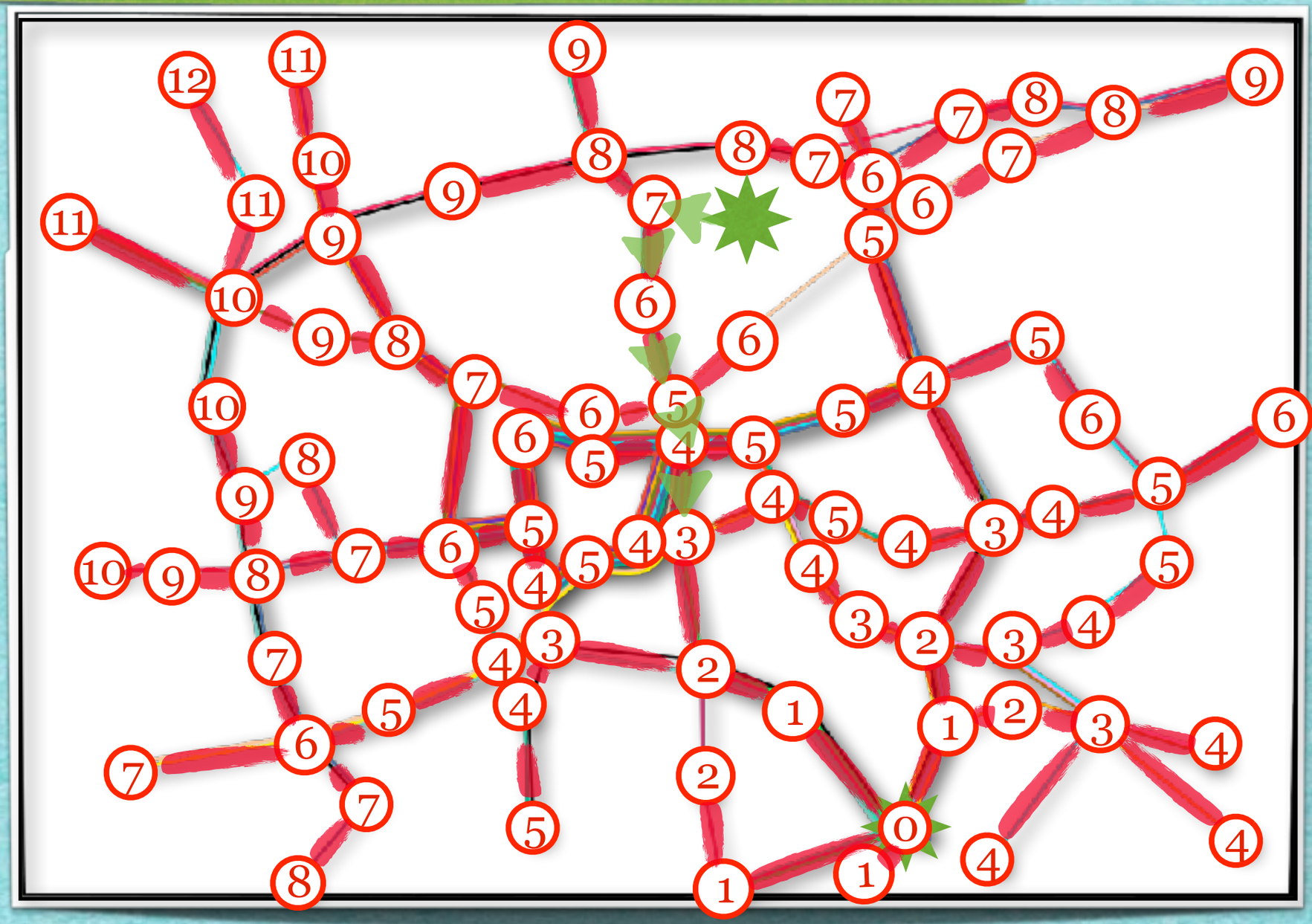
Breitensuche

Wellenreiten in Graphen



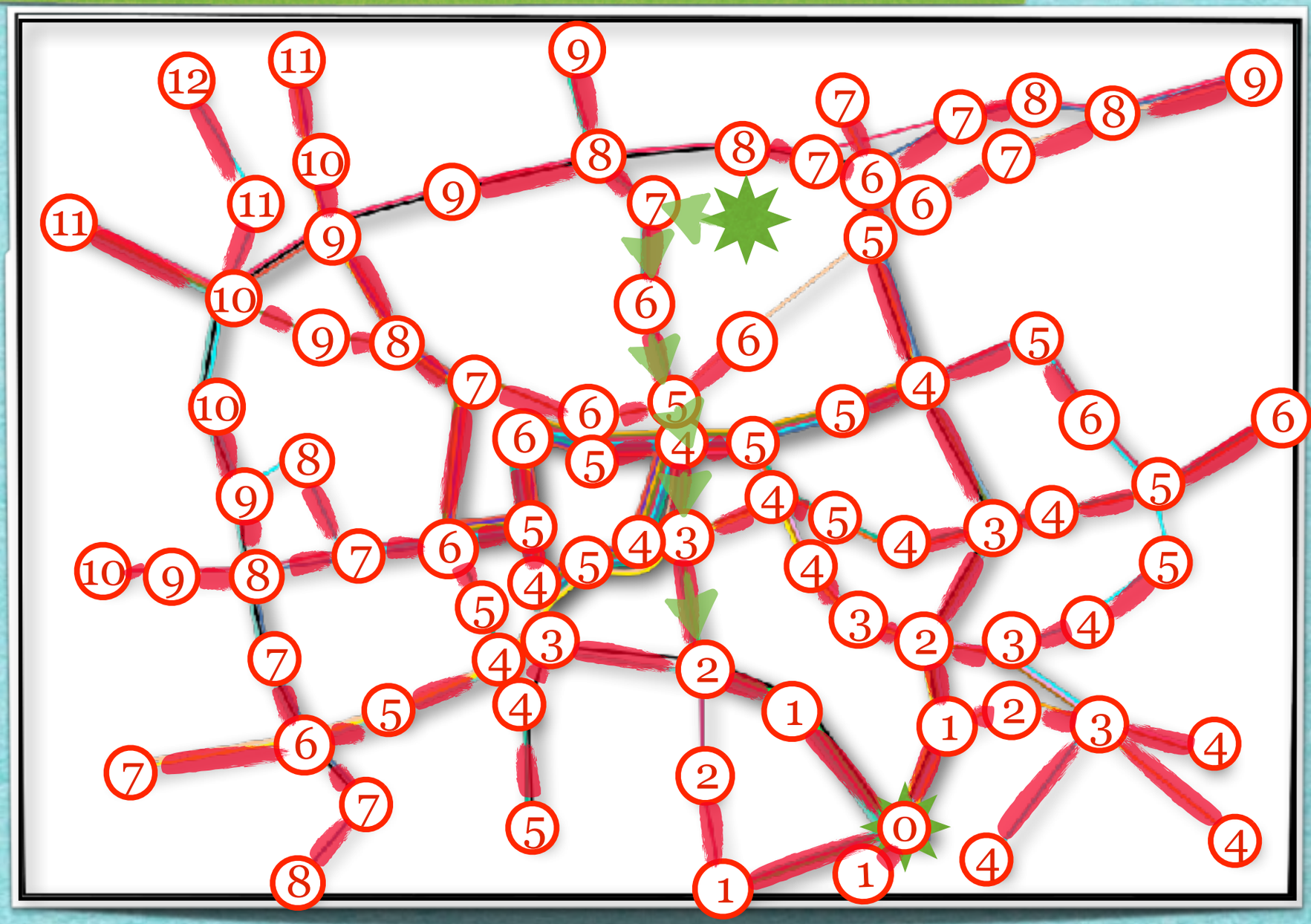
Breitensuche

Wellenreiten in Graphen



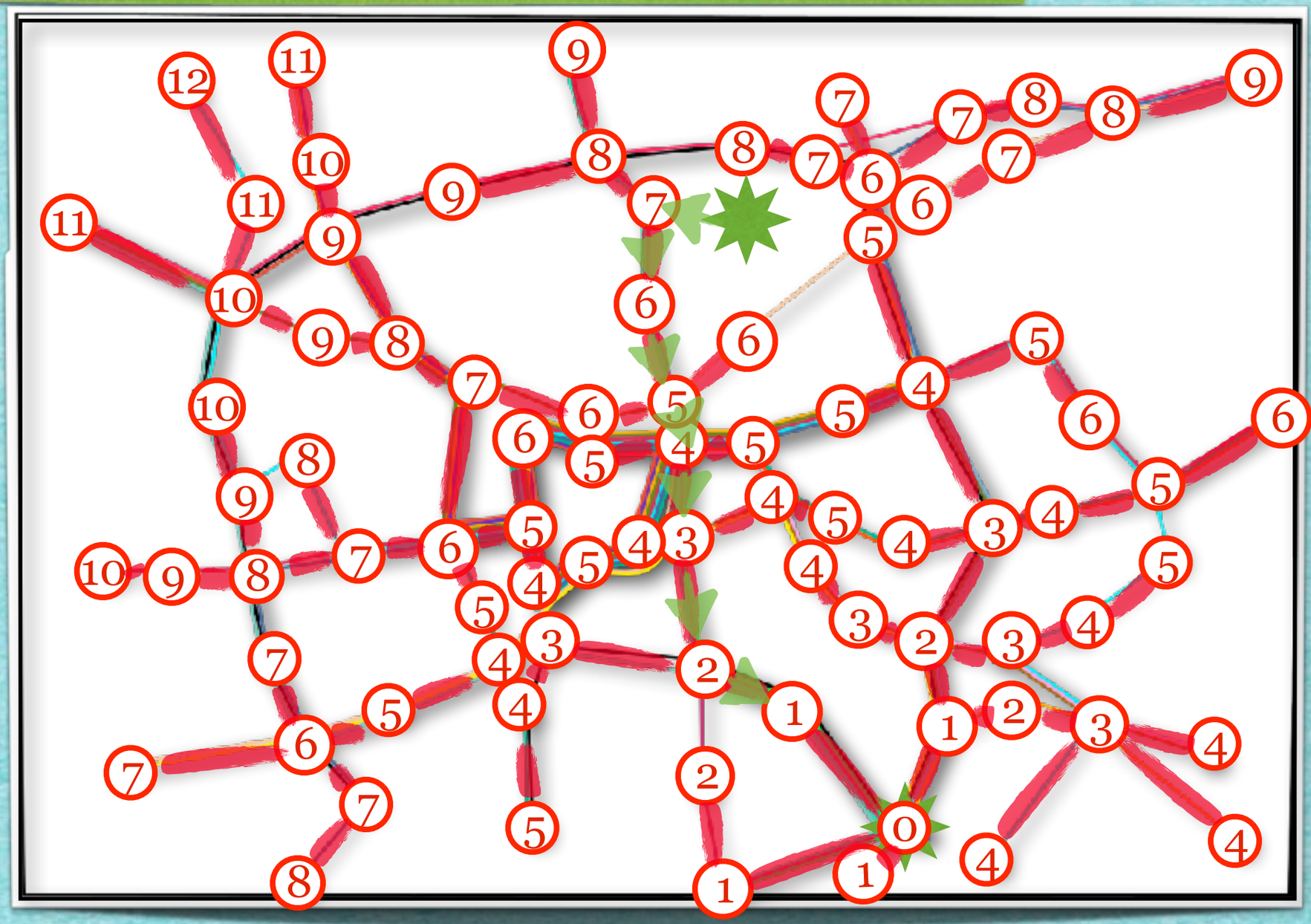
Breitensuche

Wellenreiten in Graphen



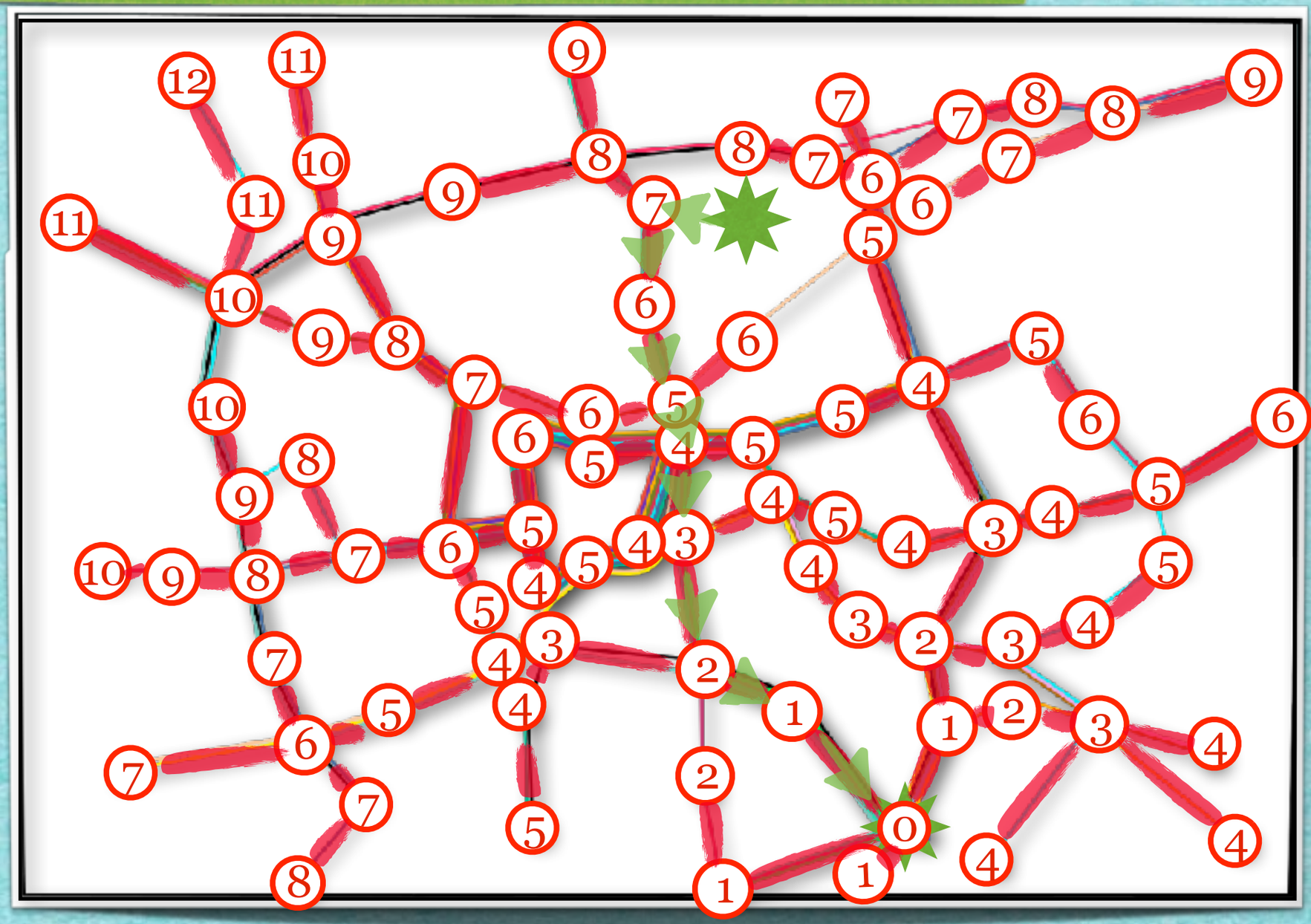
Breitensuche

Wellenreiten in Graphen



Breitensuche

Wellenreiten in Graphen



Breitensuche

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:=∅
2. WHILE (R≠∅) DO {
   - 2.1. wähle Element $v \in R$
   - 2.2. IF (es gibt kein w $V \setminus Y$ 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 u {w}, Y := Y u {w}, T := T u {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.**

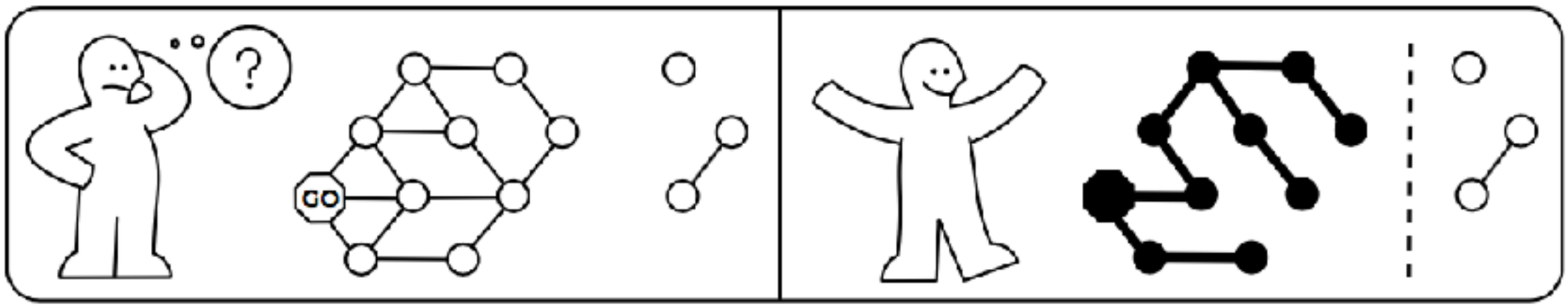
Mehr Details!

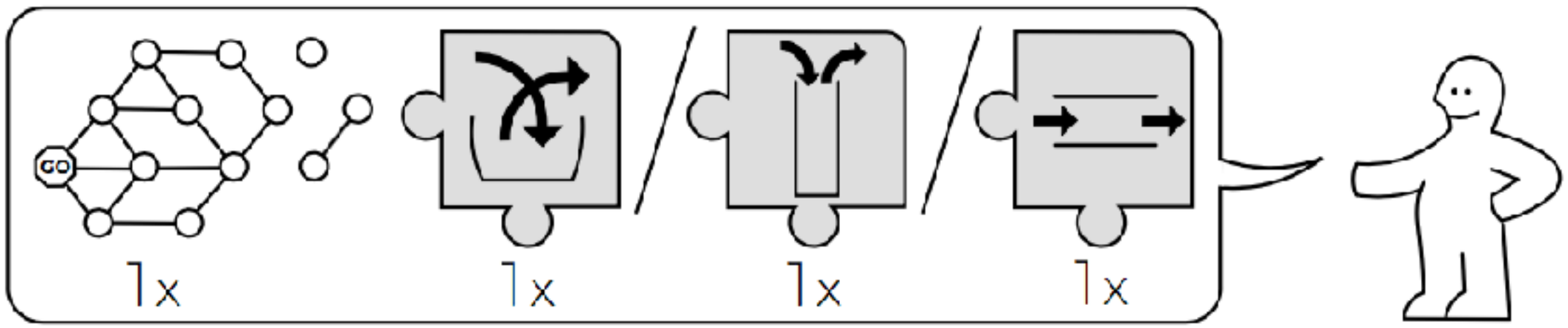
s.fekete@tu-bs.de

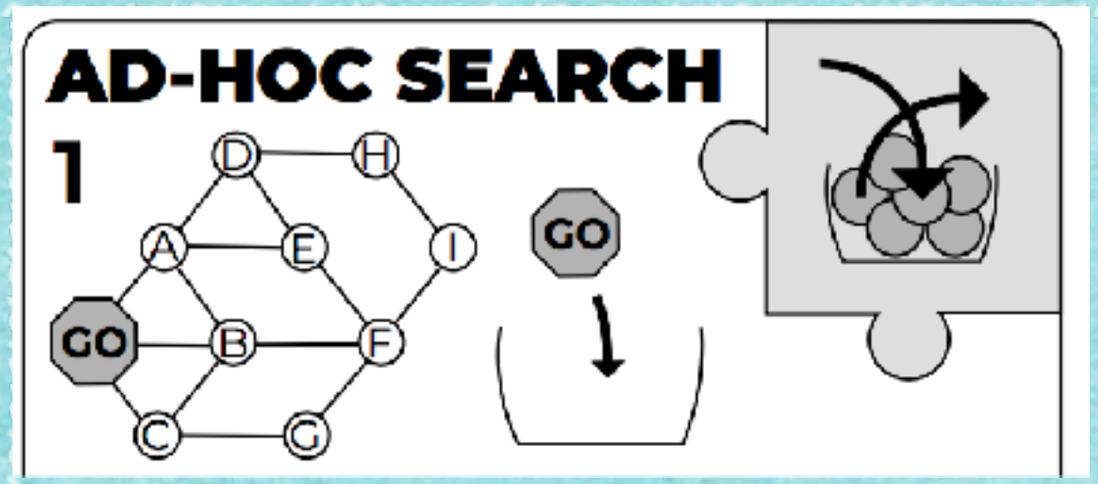
GRÅPH SCÄN

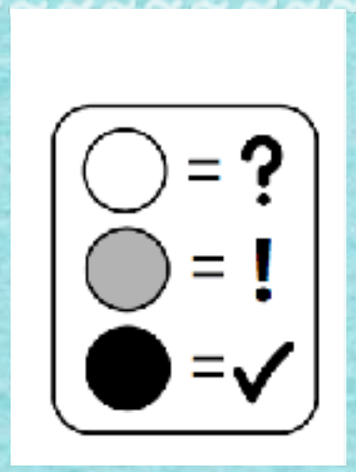
idea-instructions.com/graph-scan/
v1.0, CC by-nc-sa 4.0

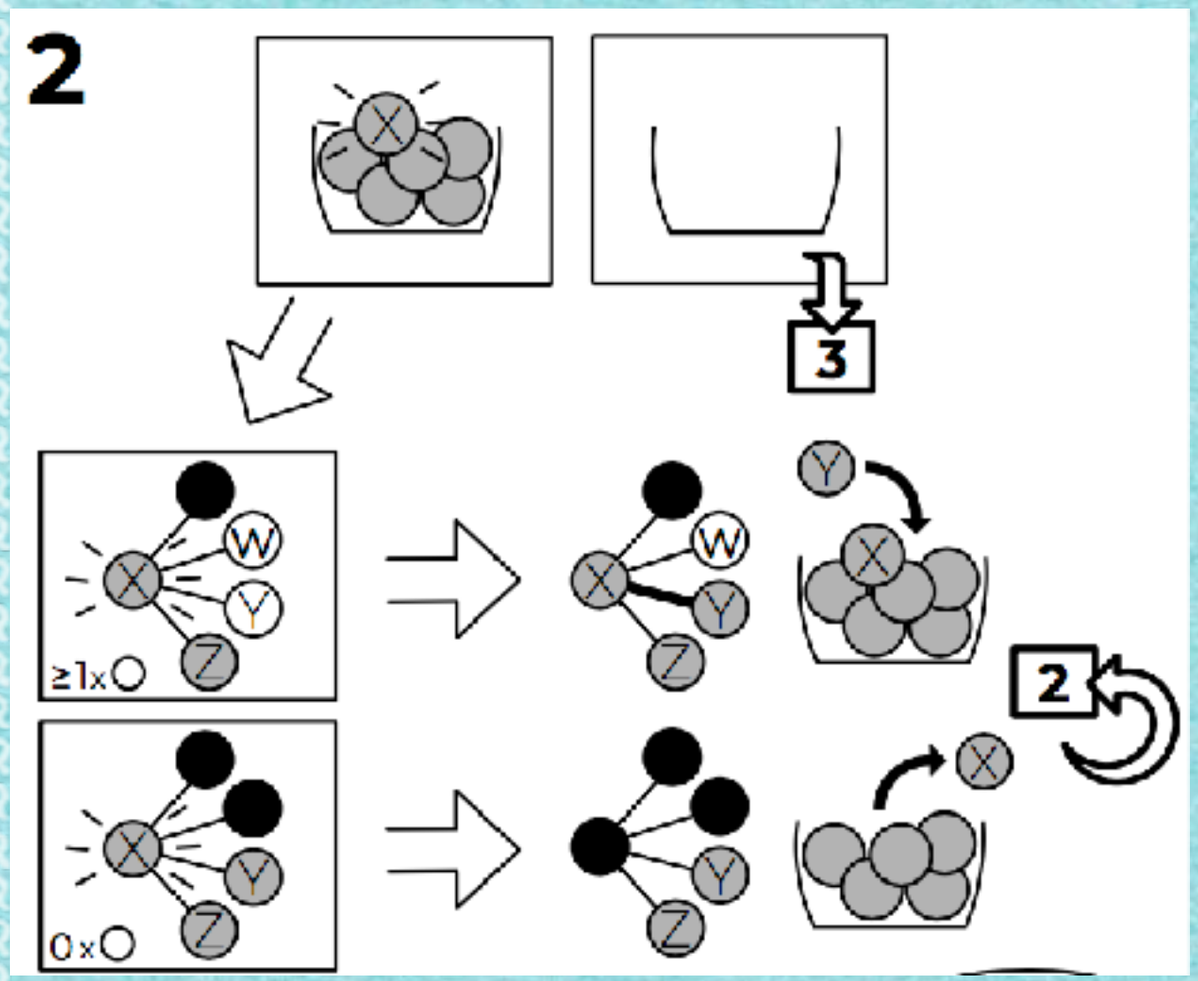
IDEA

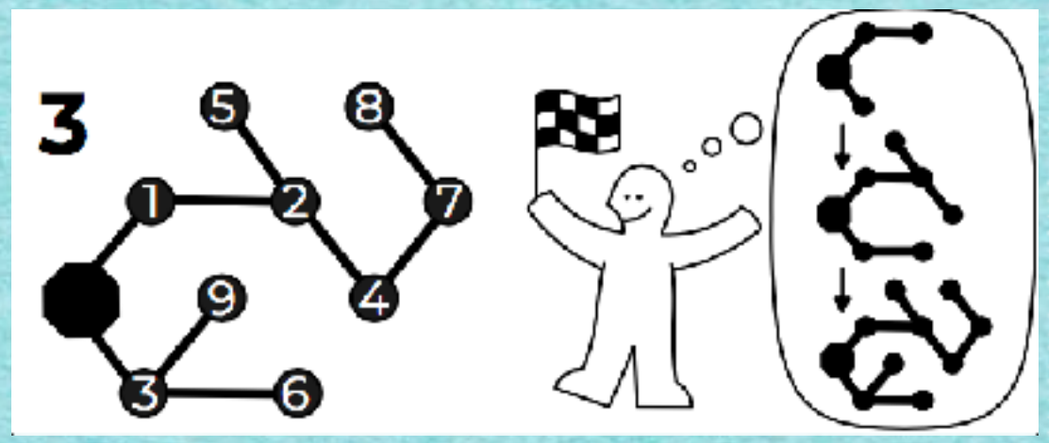


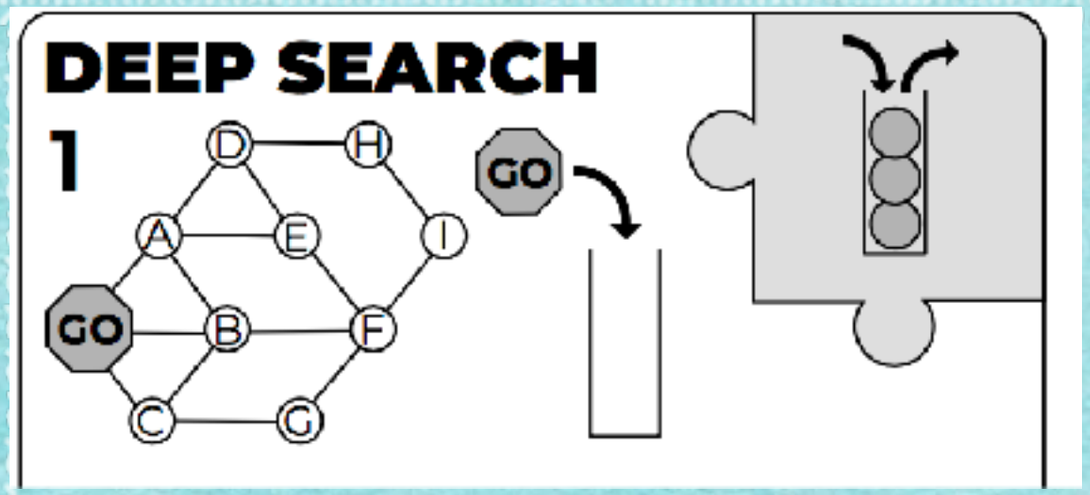


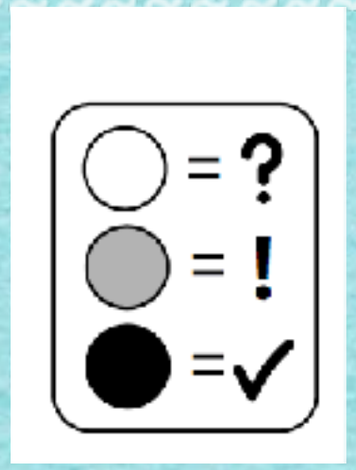

AD-HOC SEARCH
1
D
H
A
E
I
GO
B
F
C
G
GO

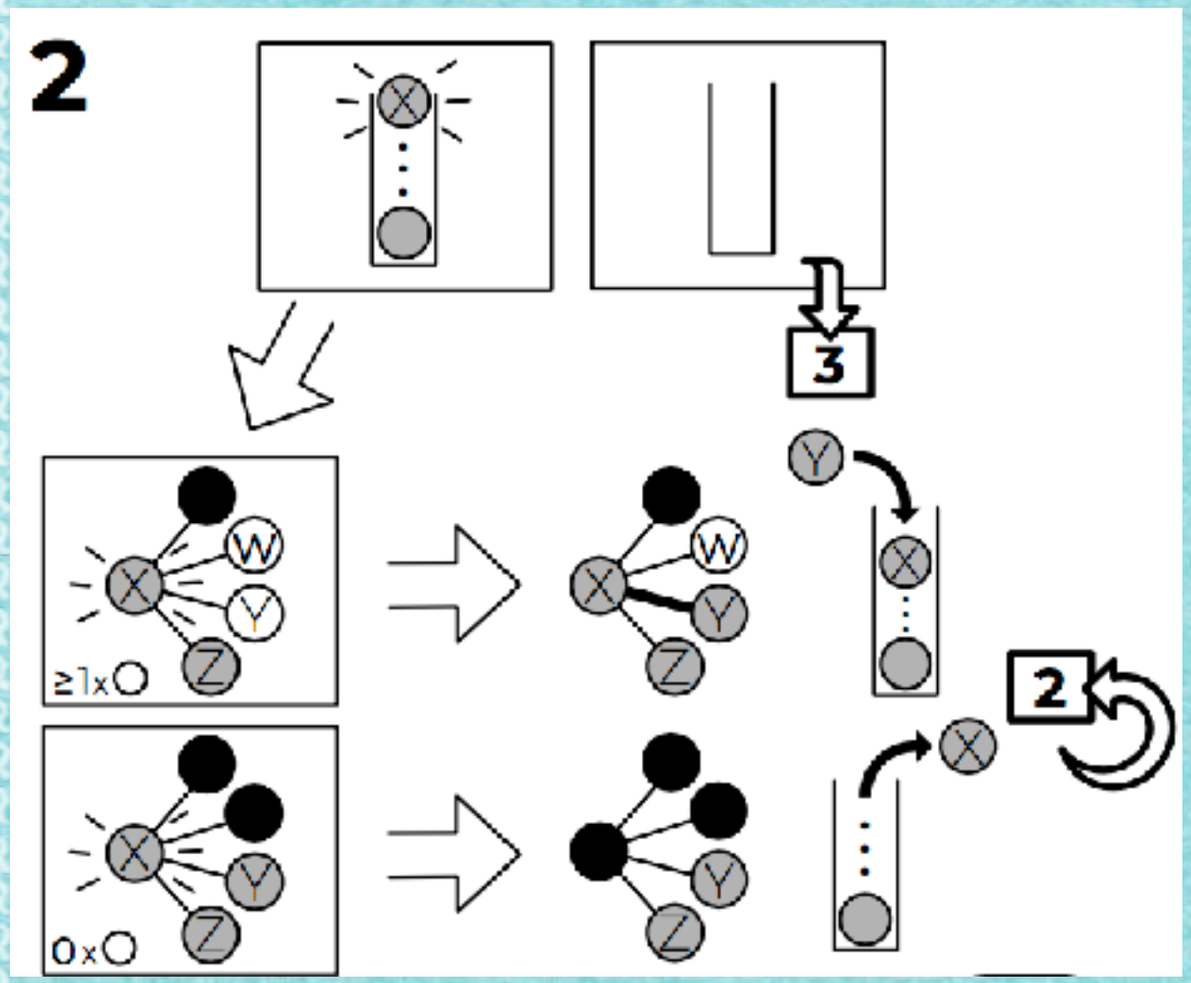

= ?
= !
= ✓

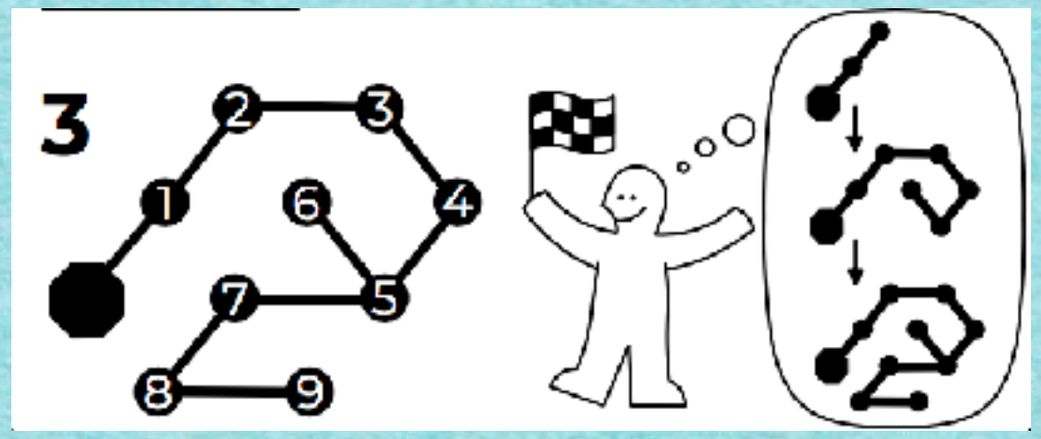

2
X
3
W
X
Y
Z
≥1x ○
0x ○
2


3
5
8
1
2
7
9
4
3
6

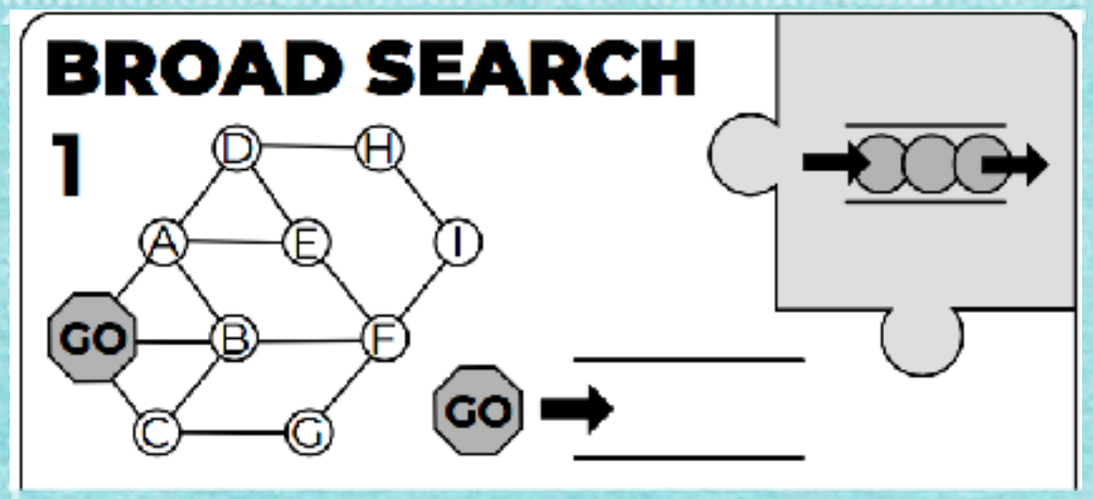

DEEP SEARCH
1
D
H
GO
A
E
I
GO
B
F
C
G

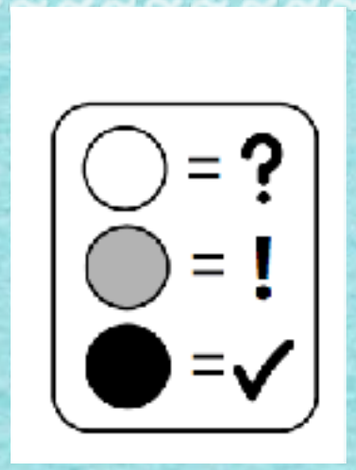

= ?
= !
=

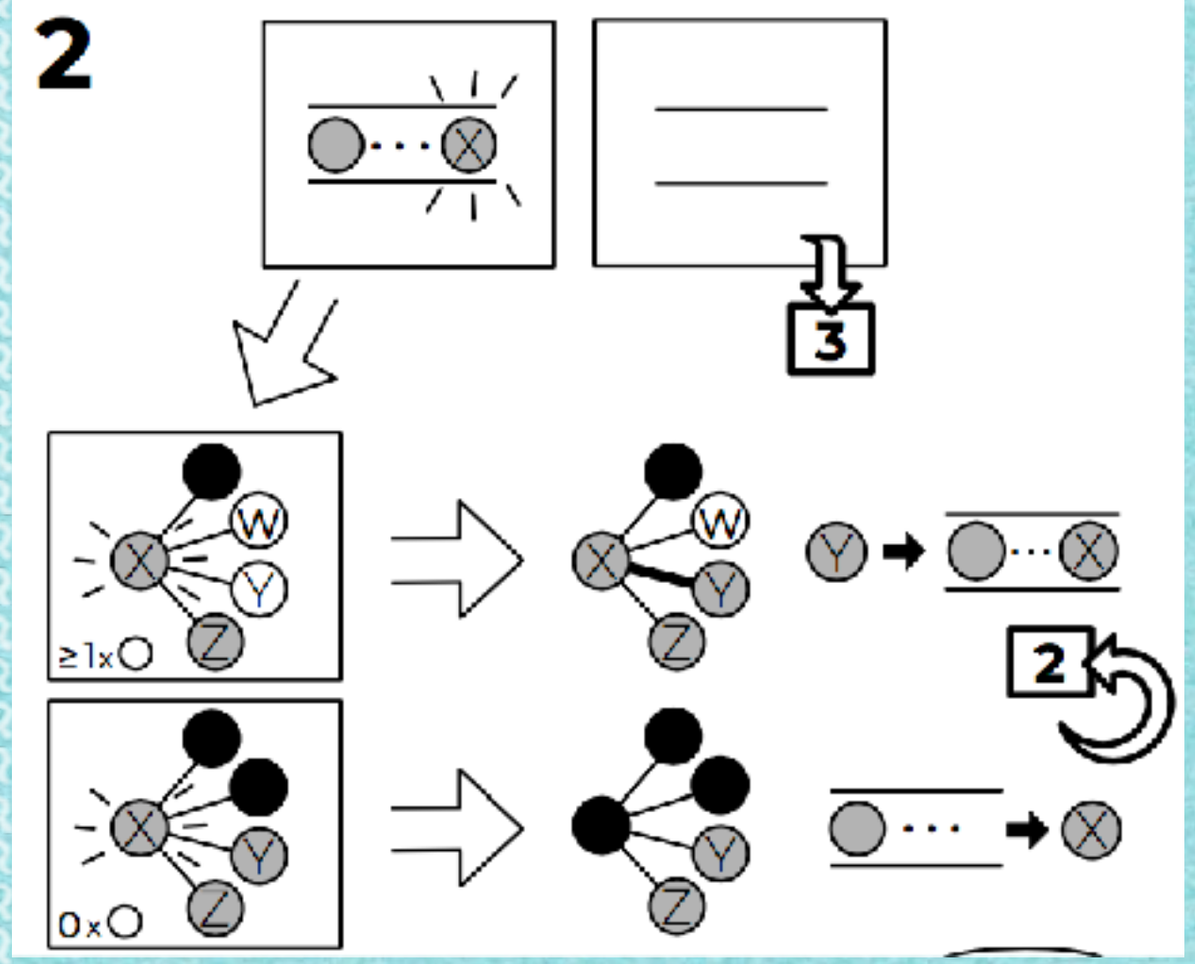

2
3
W
X
Y
Z
≥1x
0x
2

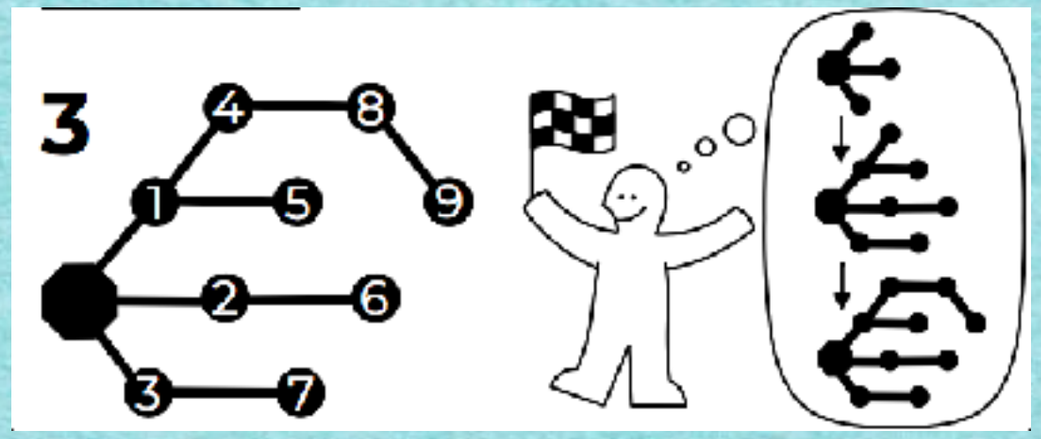

3
1
2
3
4
5
6
7
8
9

BROAD SEARCH





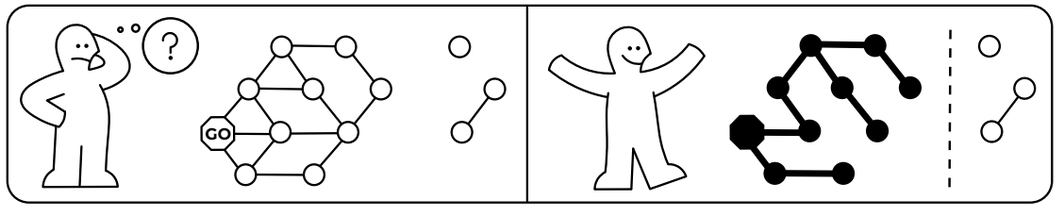


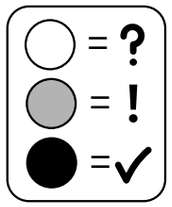


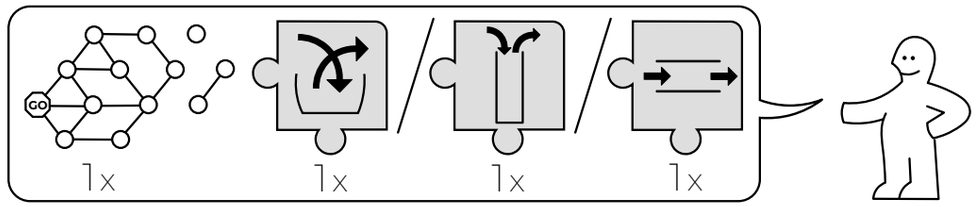
GRÅPH SCÄN

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

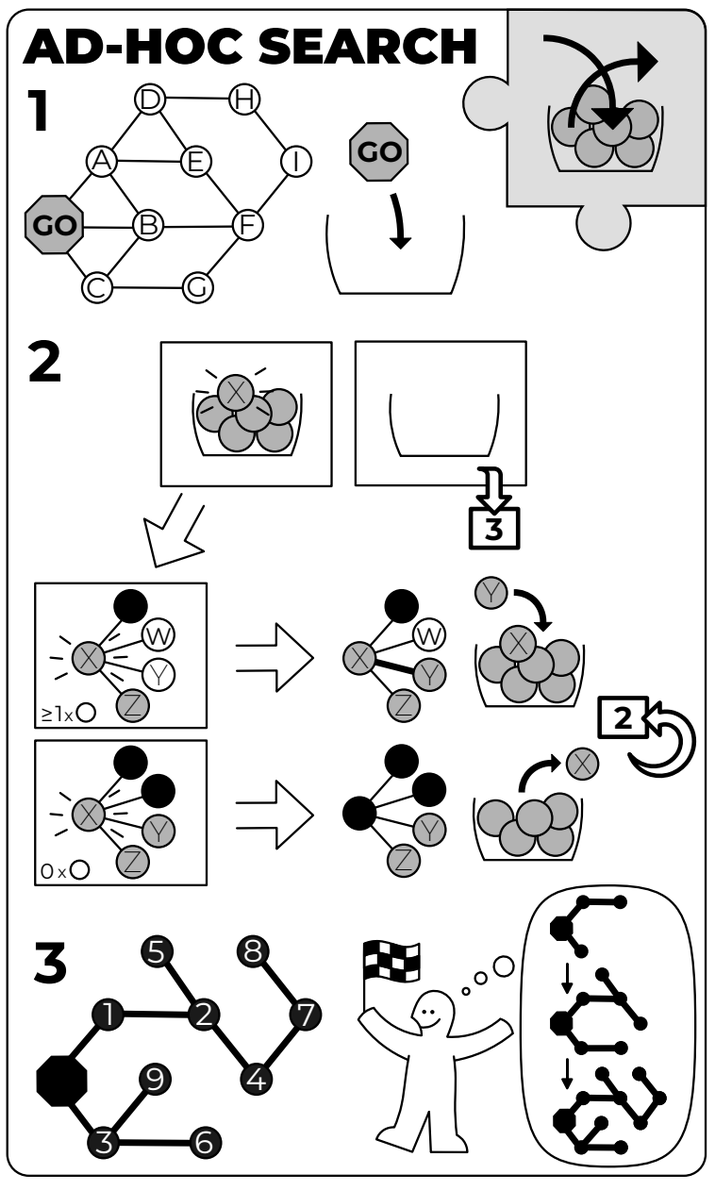
IDEA



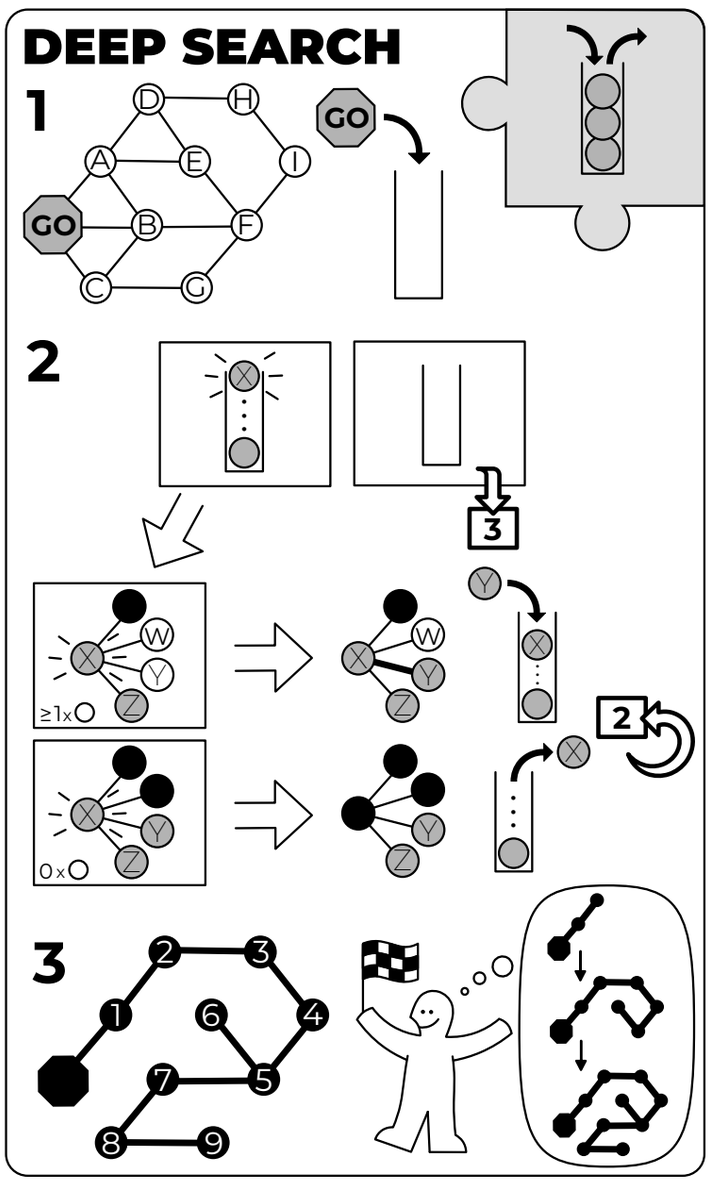




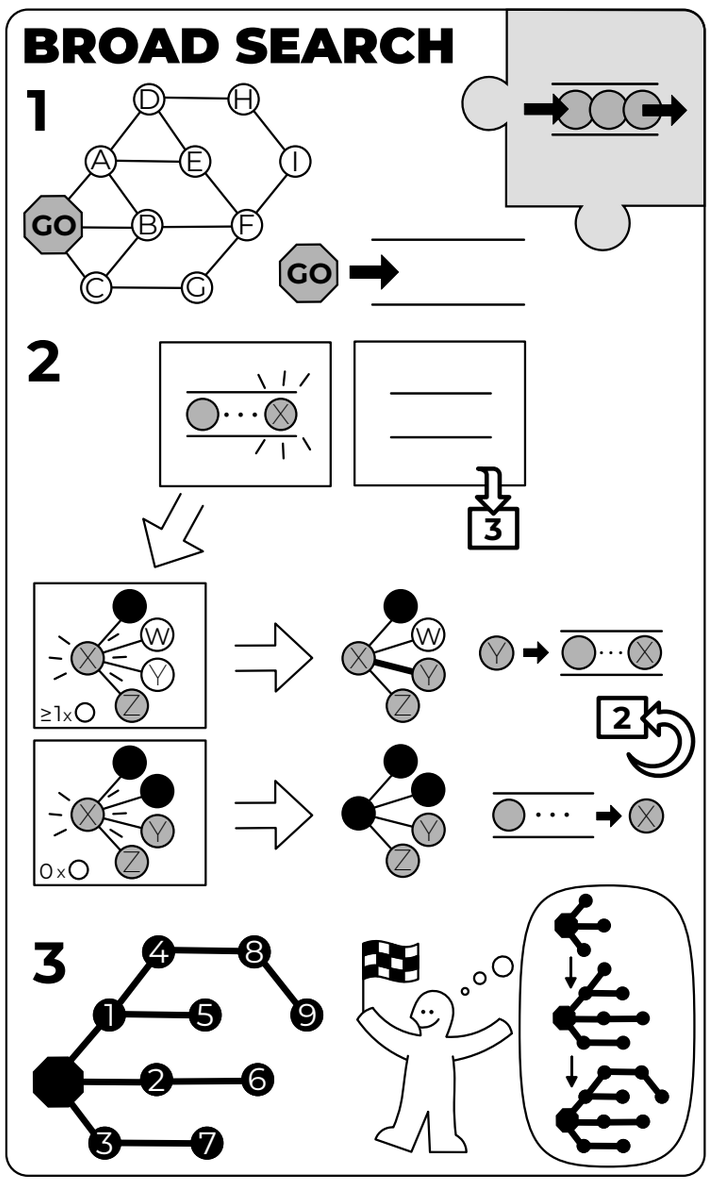
AD-HOC SEARCH



DEEP SEARCH



BROAD SEARCH



Kapitelende!

s.fekete@tu-bs.de