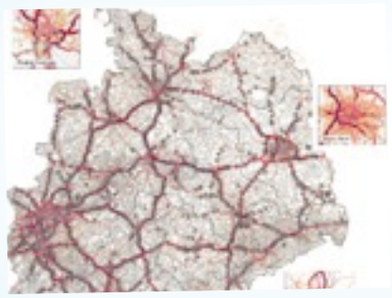


Auf die Schnelle mit der Welle: Wie findet man kürzeste Wege?

Kürzeste Wege, Graphen und Algorithmen

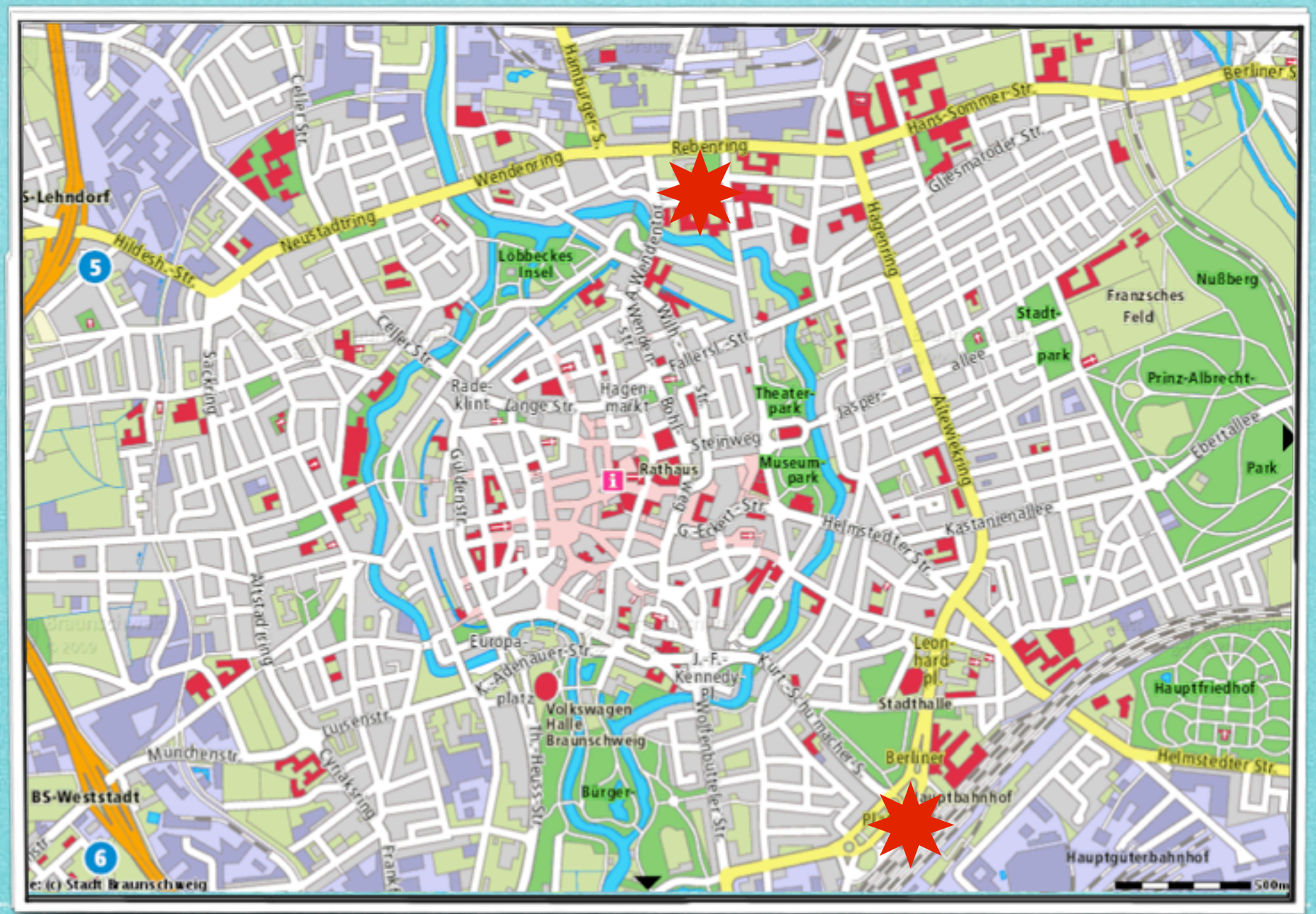
Prof. Dr. Sándor Fekete

Wie kommt man zur TU?

Wie kommt man zur TU?

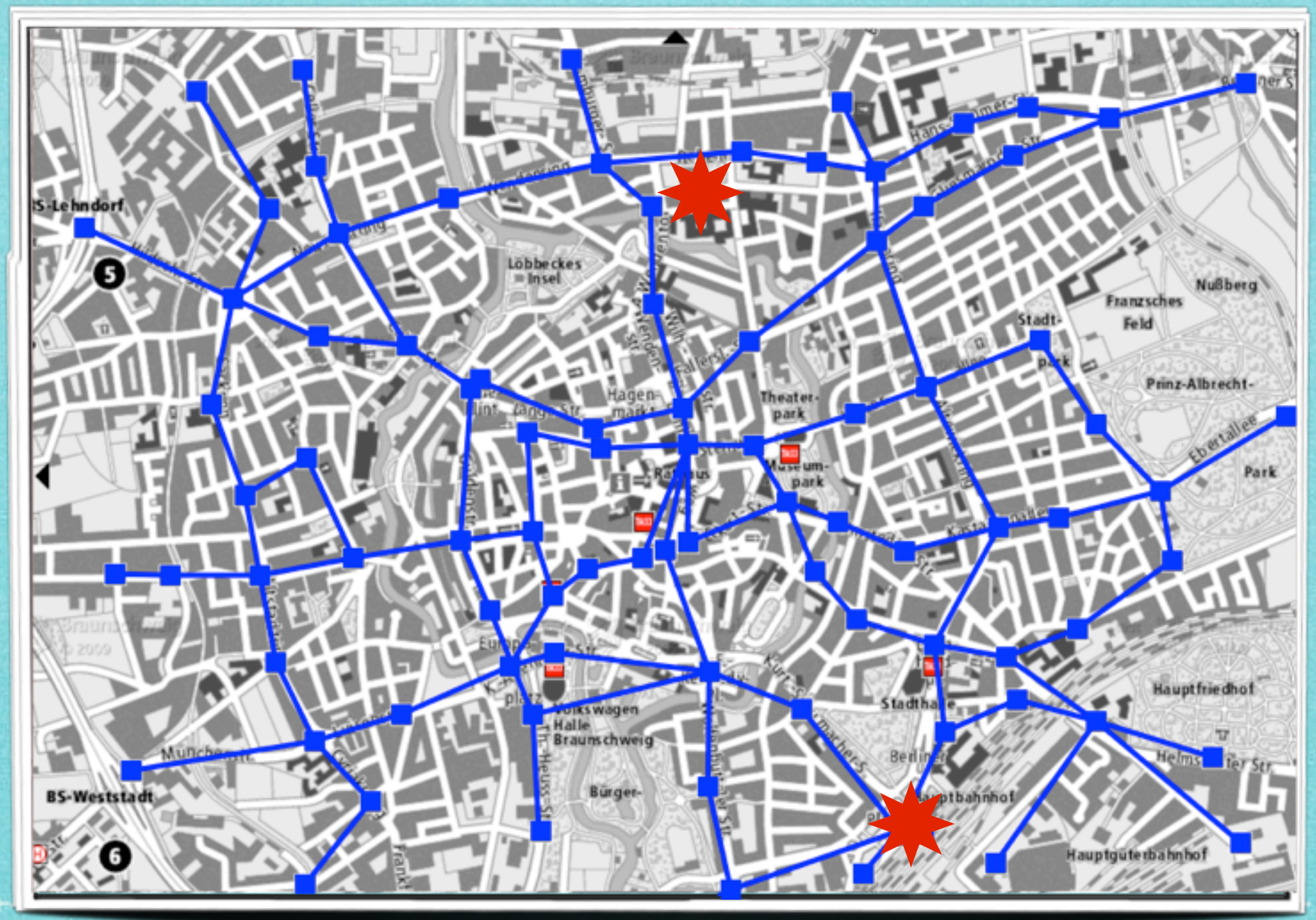
Wie kommt man zur TU?

Wie kommt man zur TU?



Wie kommt man zur TU?

Wie kommt man zur TU?



Auf die Schnelle mit der Welle

Auf die Schnelle mit der Welle

Auf die Schnelle mit der Welle

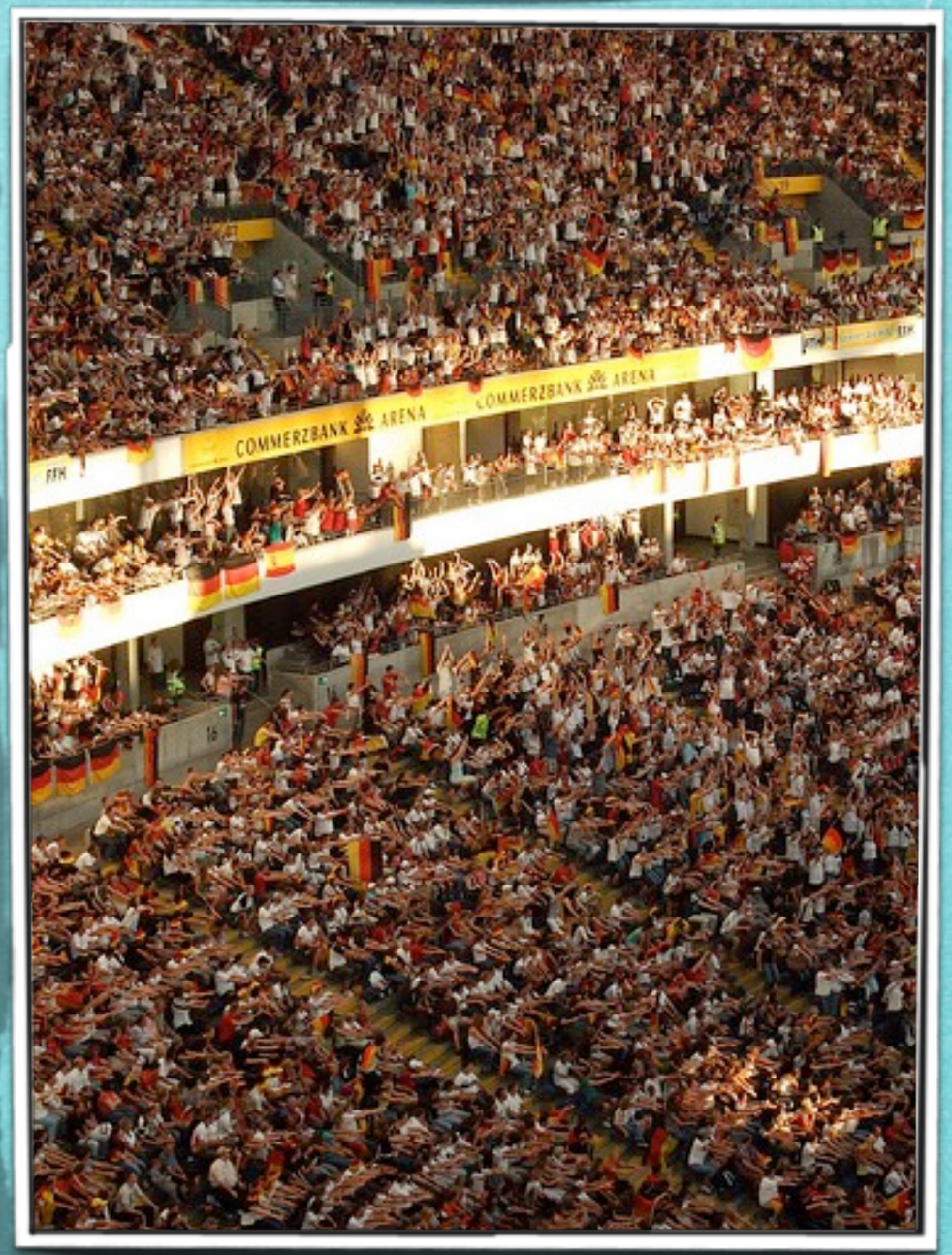
A. LOS bei „NULL“

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

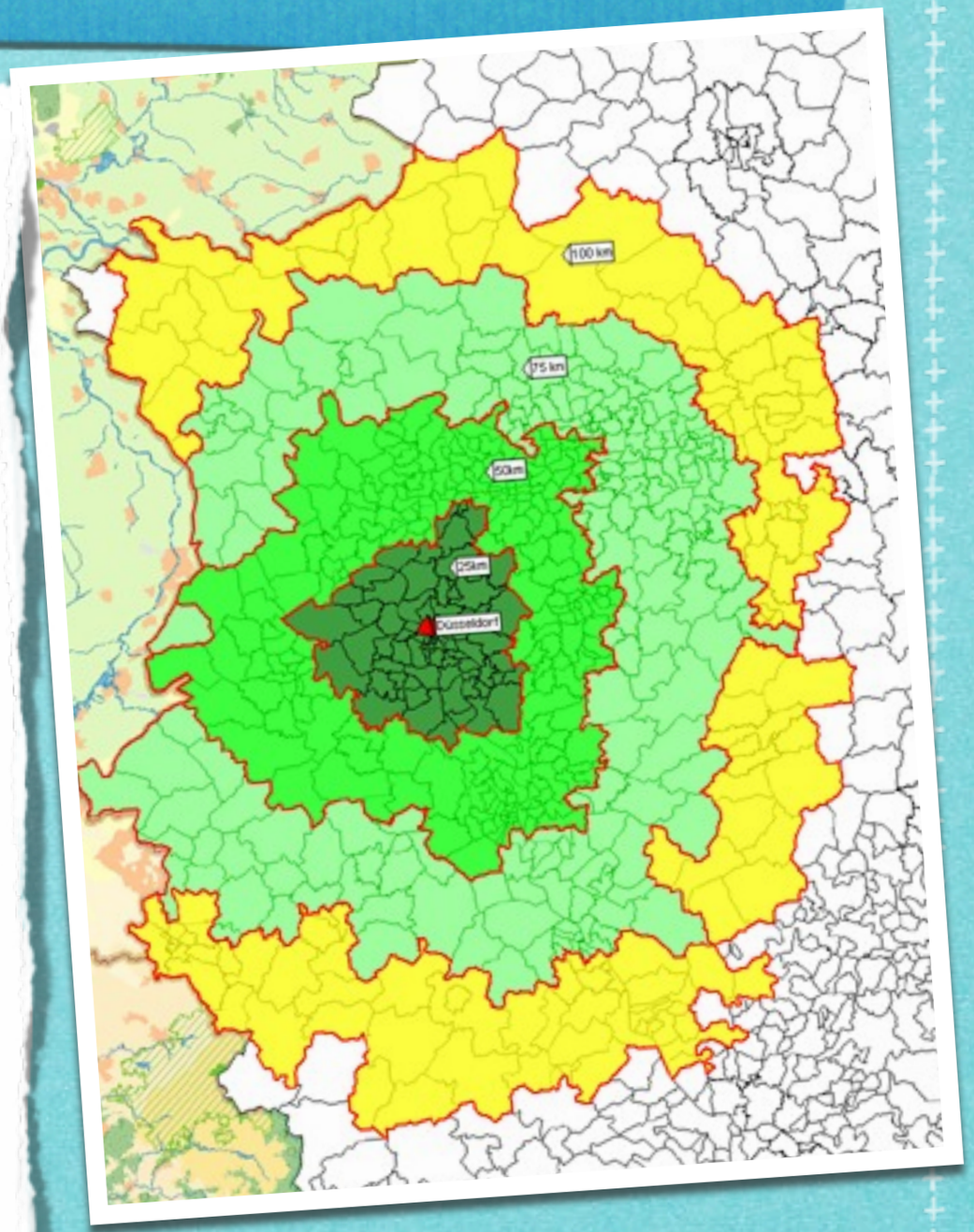
A. LOS bei „NULL“

B. Bis „ANGEKOMMEN!“:

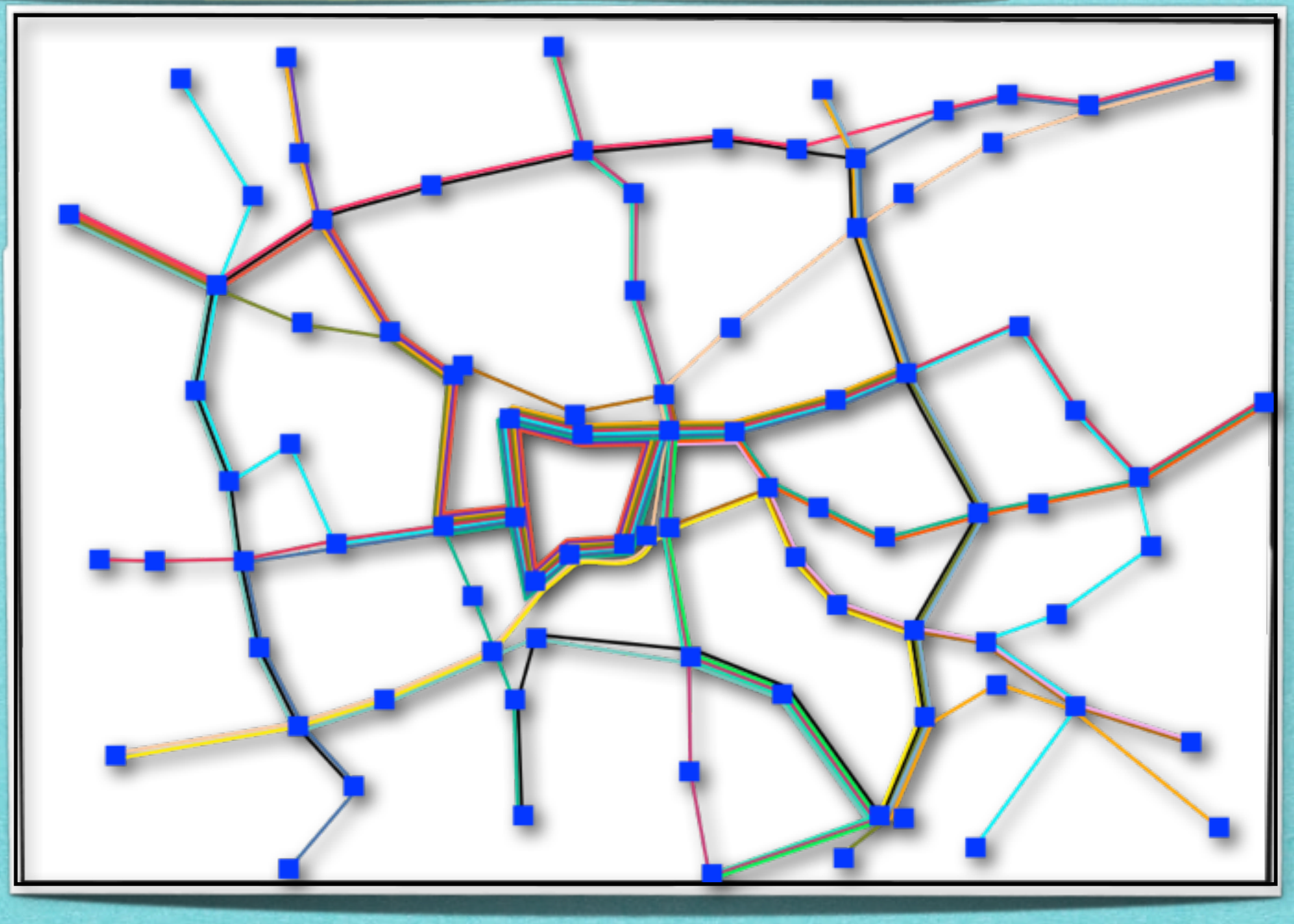
- Solange du noch nicht aufgestanden warst:
  - Wenn ein oder mehrere direkte Nachbarn aufstehen:
    1. Einen dieser Nachbarn merken
    2. In der nächsten Runde:
       2.1. aufstehen
       2.2. Zahl merken
    3. In der übernächsten Runde hinsetzen

C. Nach „ANGEKOMMEN!“:

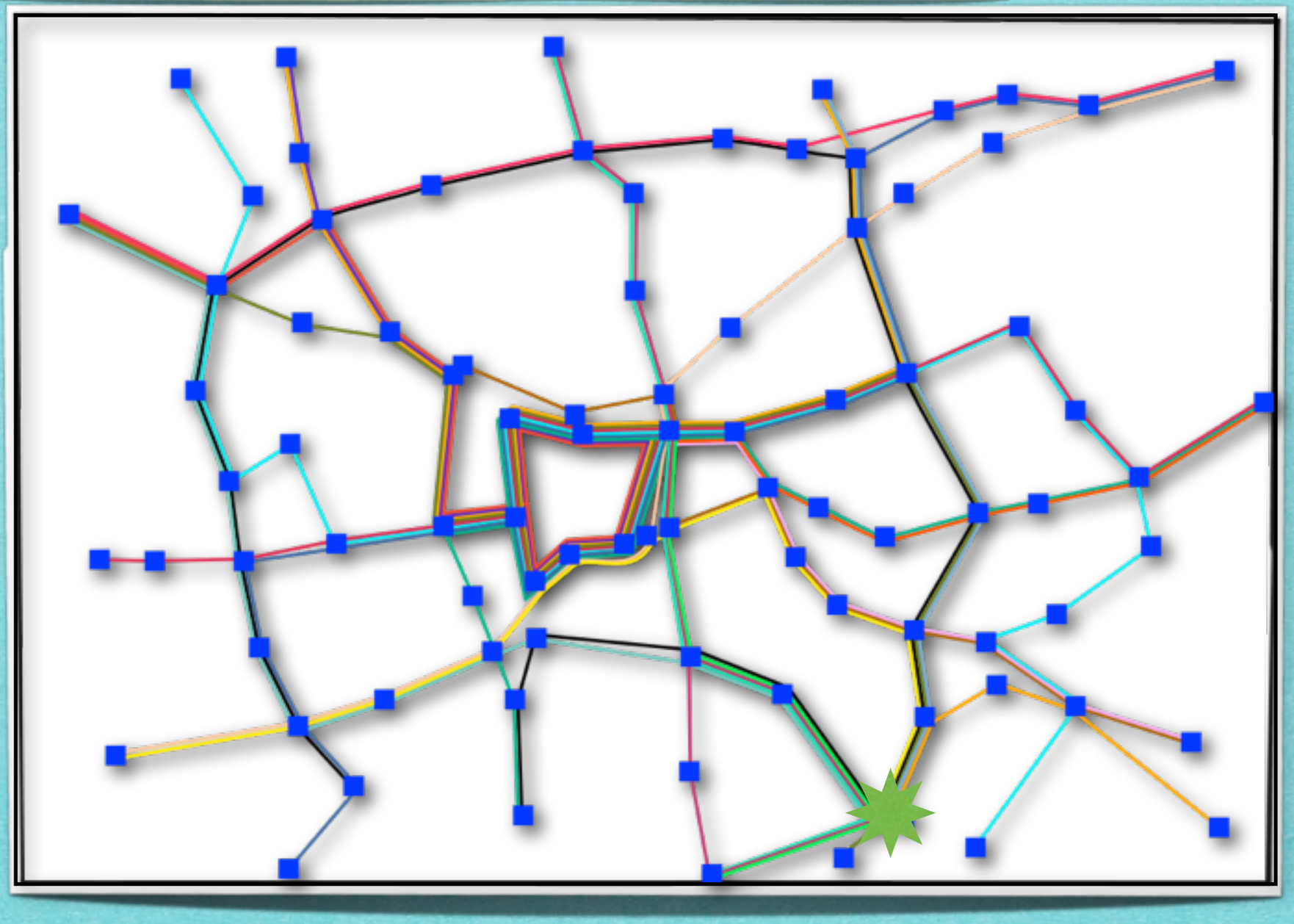
- Auf gemerkten Nachbarn zeigen



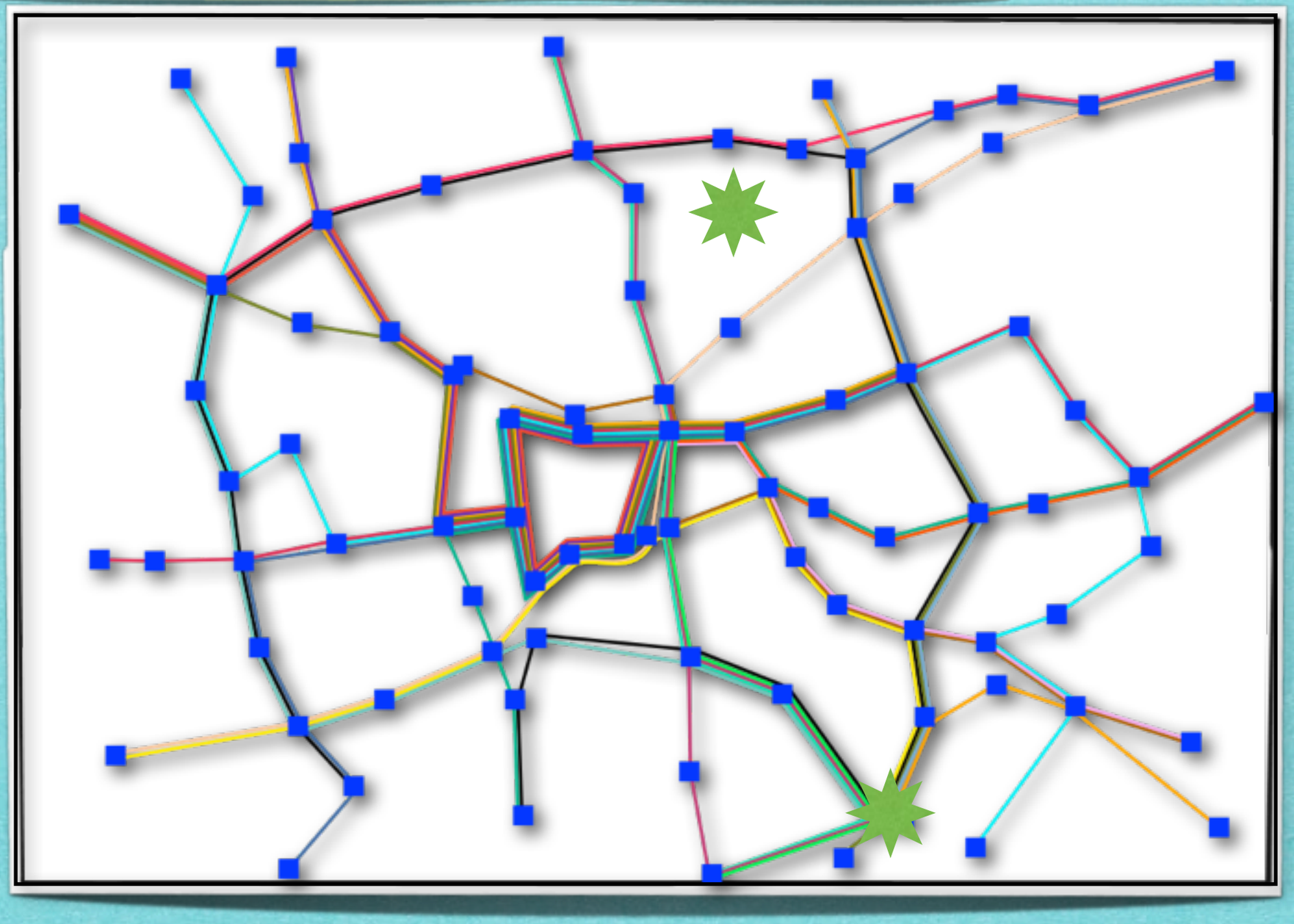
Wellenreiten in Graphen



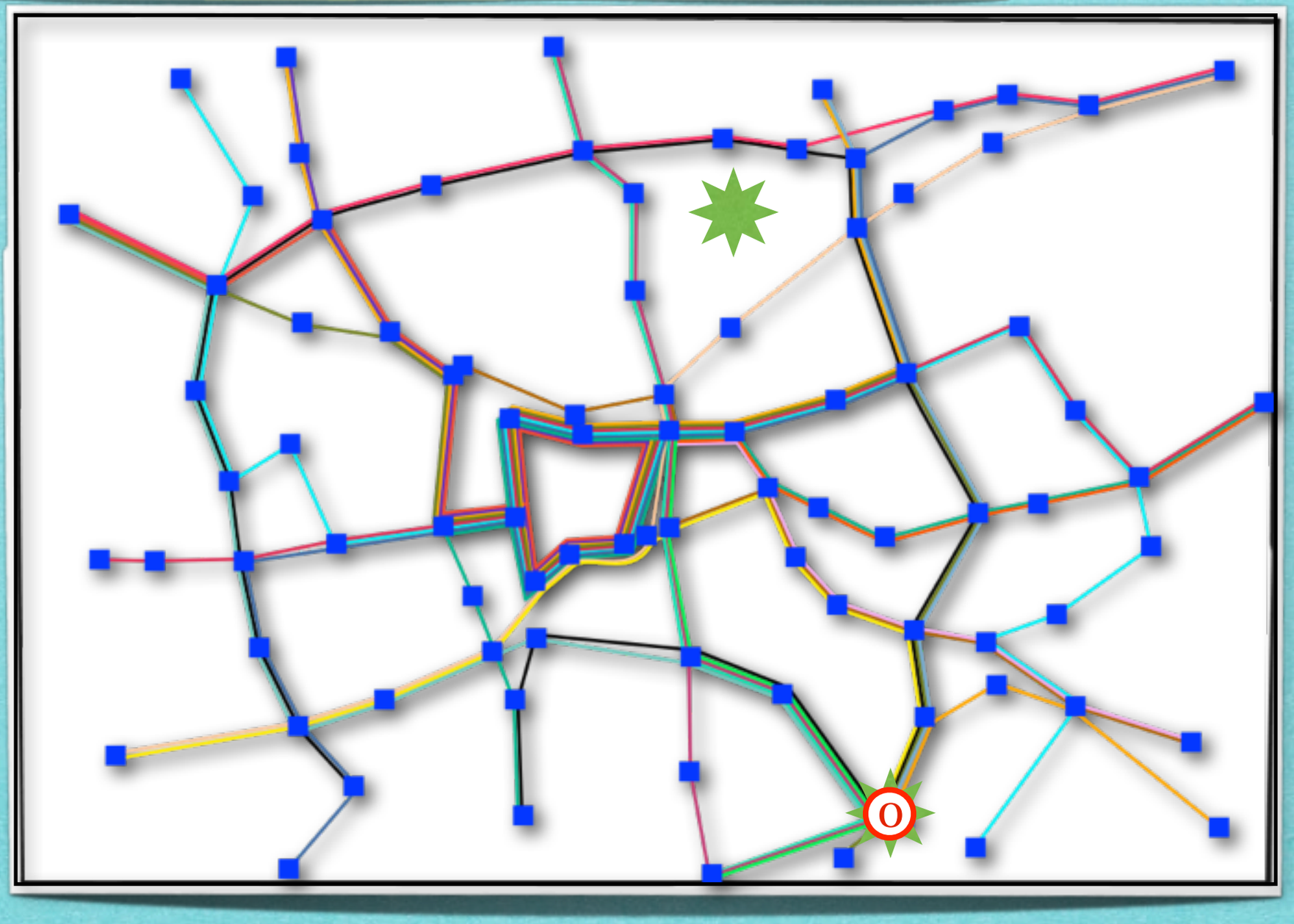
Wellenreiten in Graphen



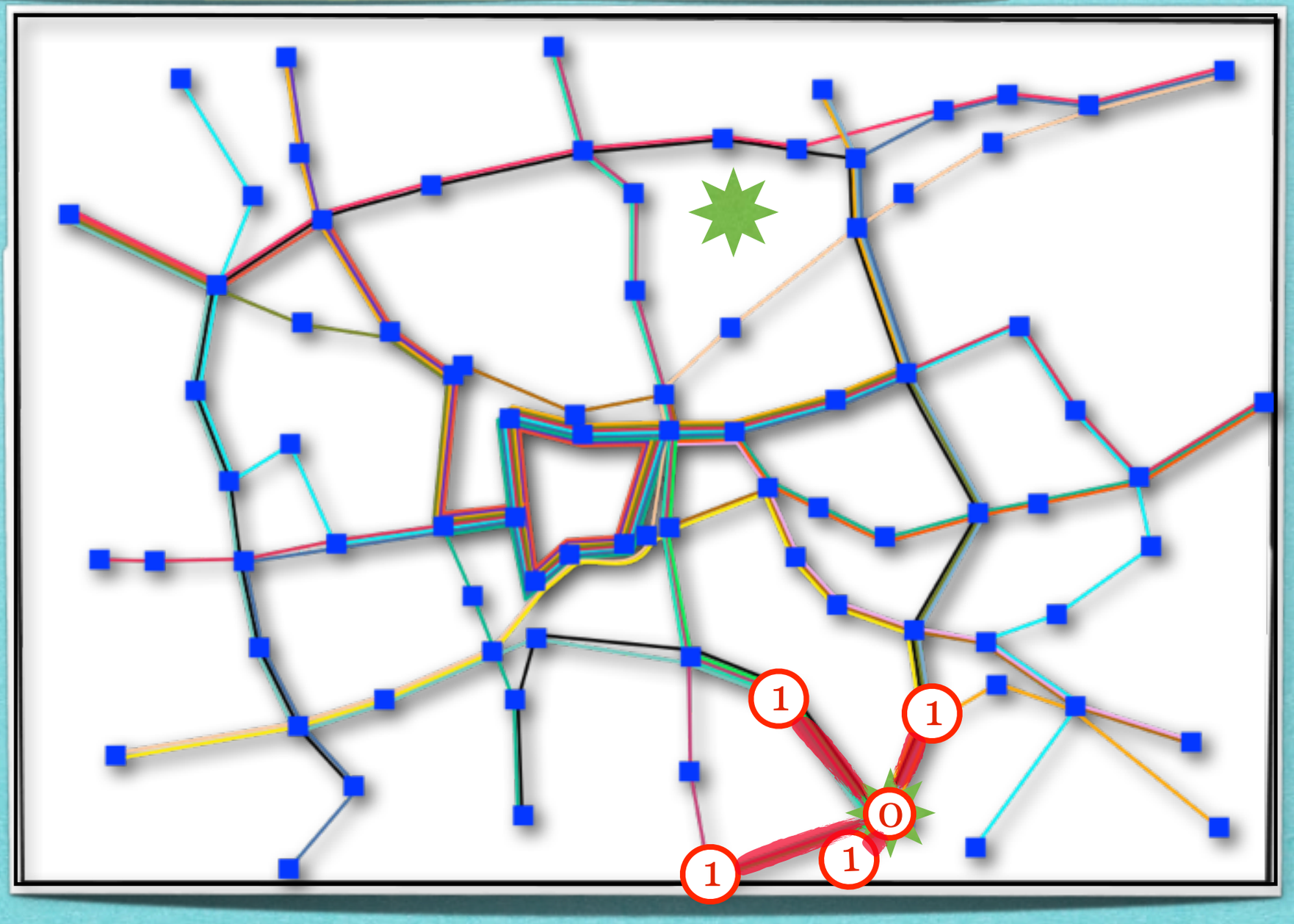
Wellenreiten in Graphen



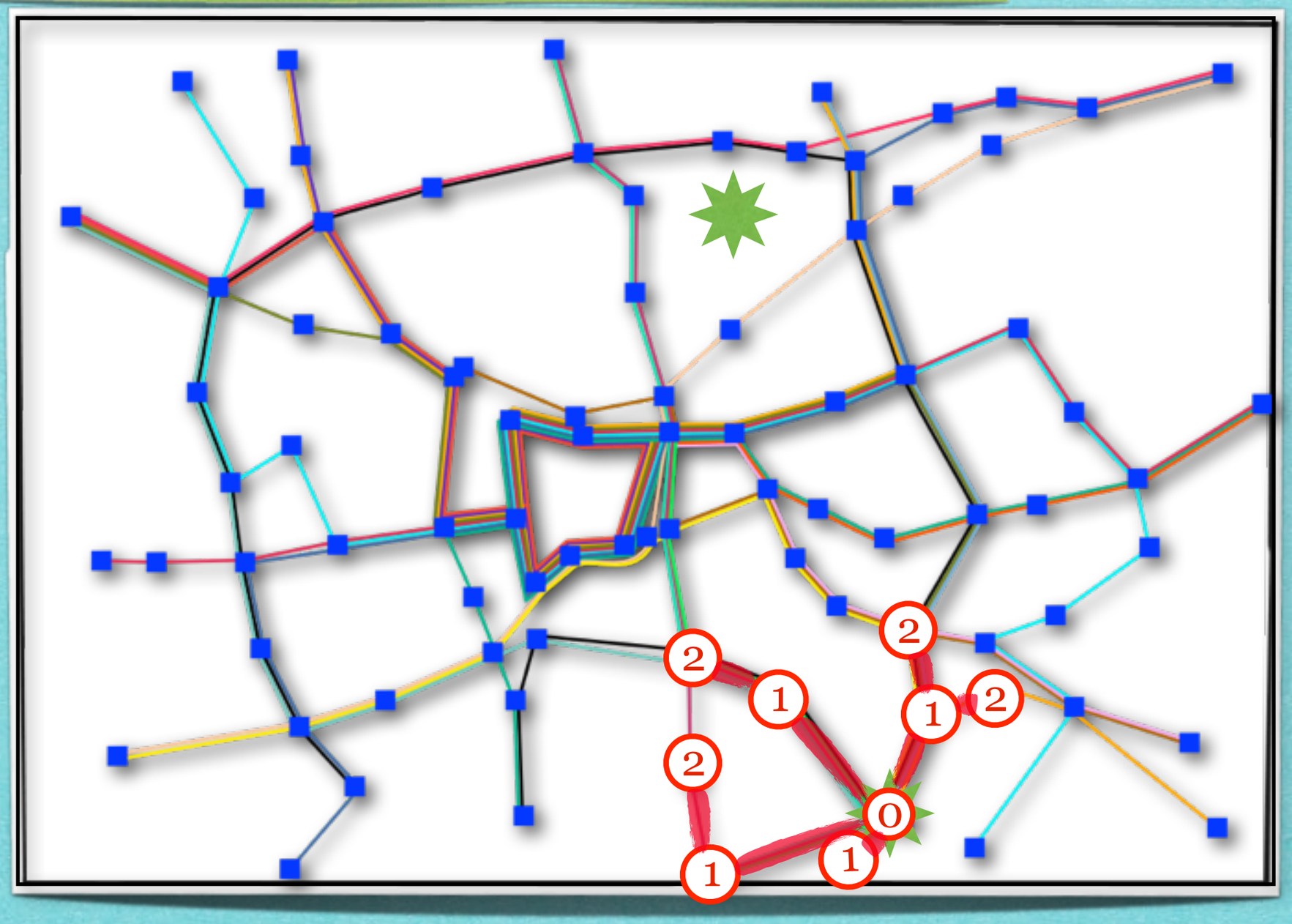
Wellenreiten in Graphen



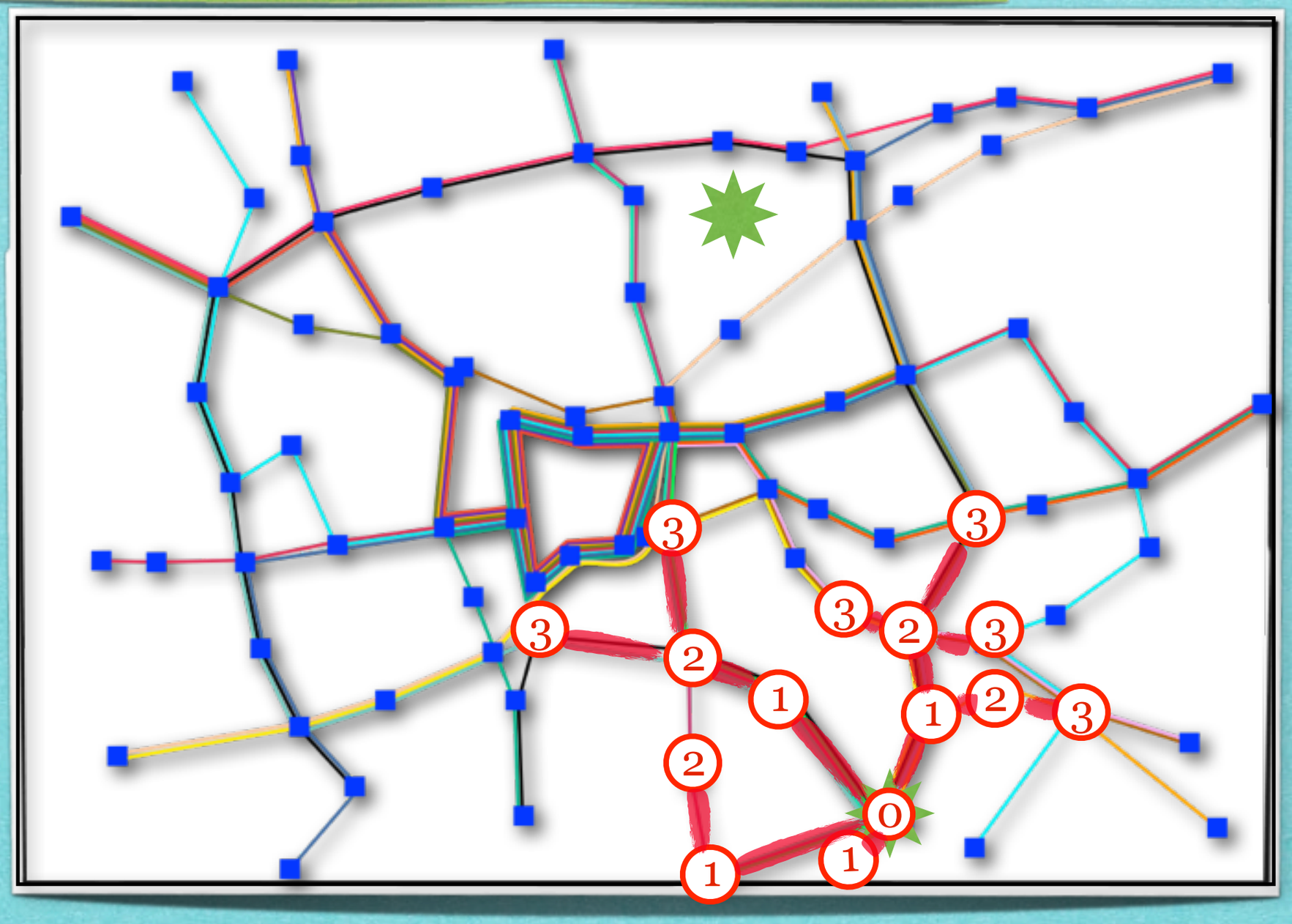
Wellenreiten in Graphen



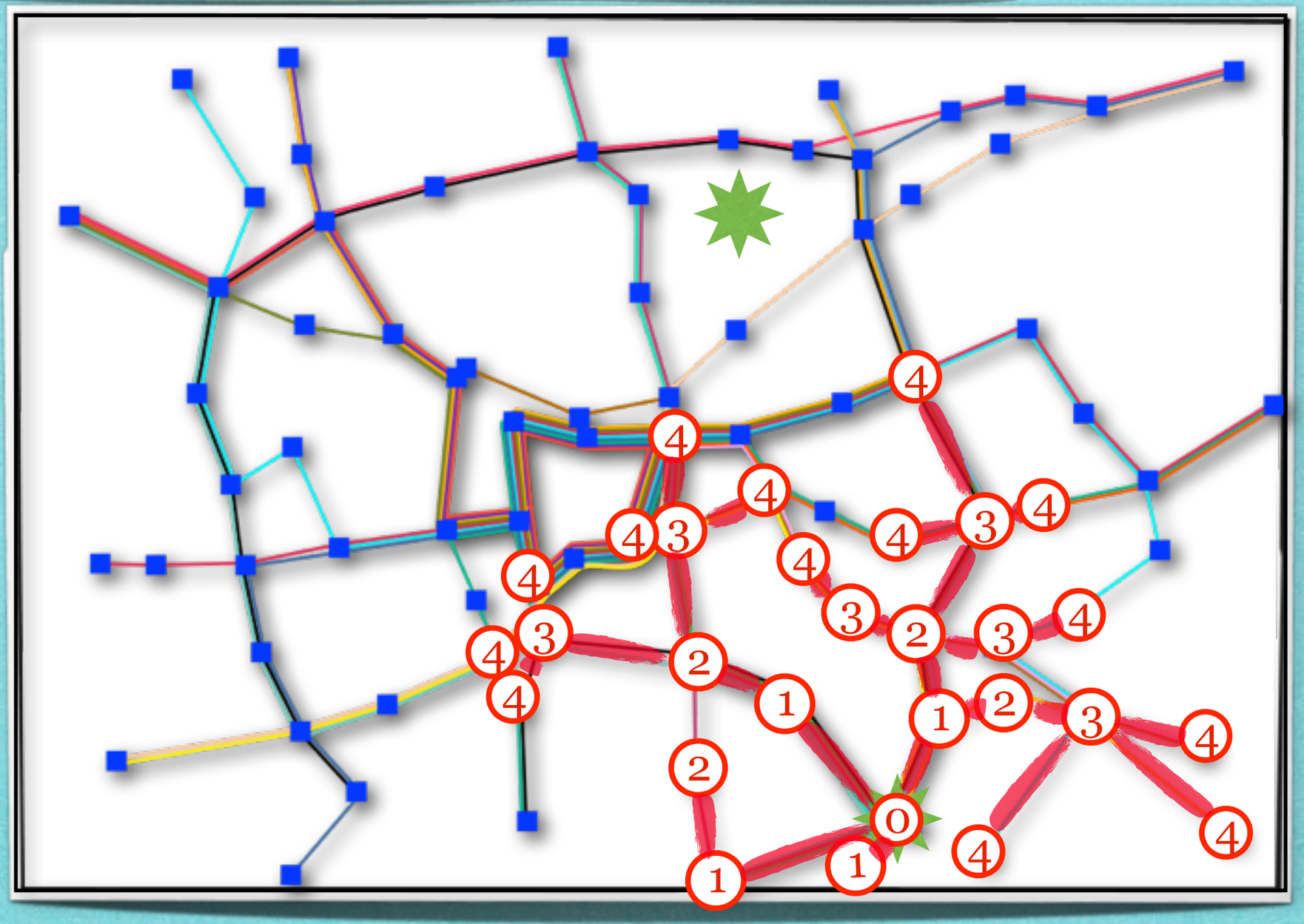
Wellenreiten in Graphen



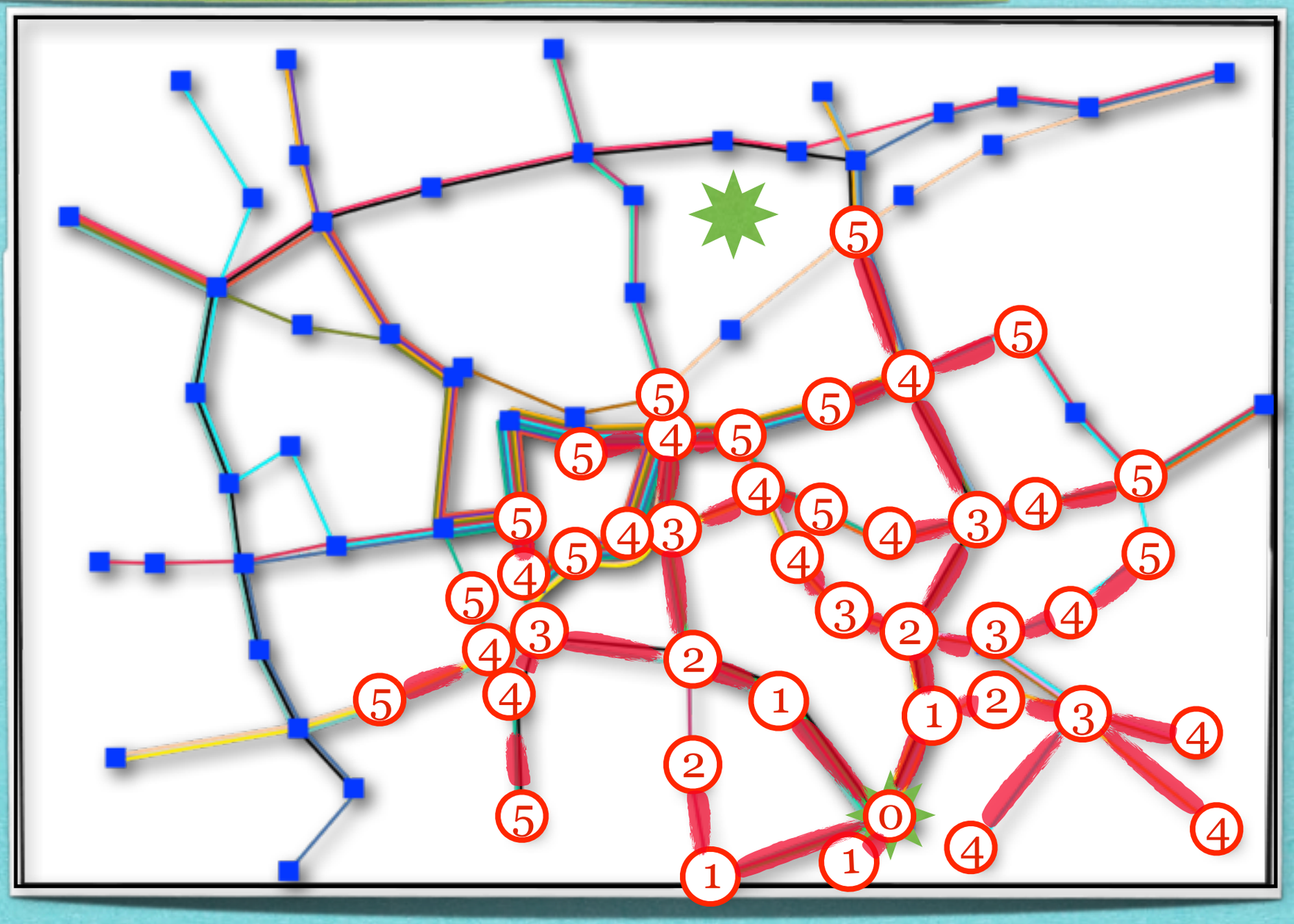
Wellenreiten in Graphen



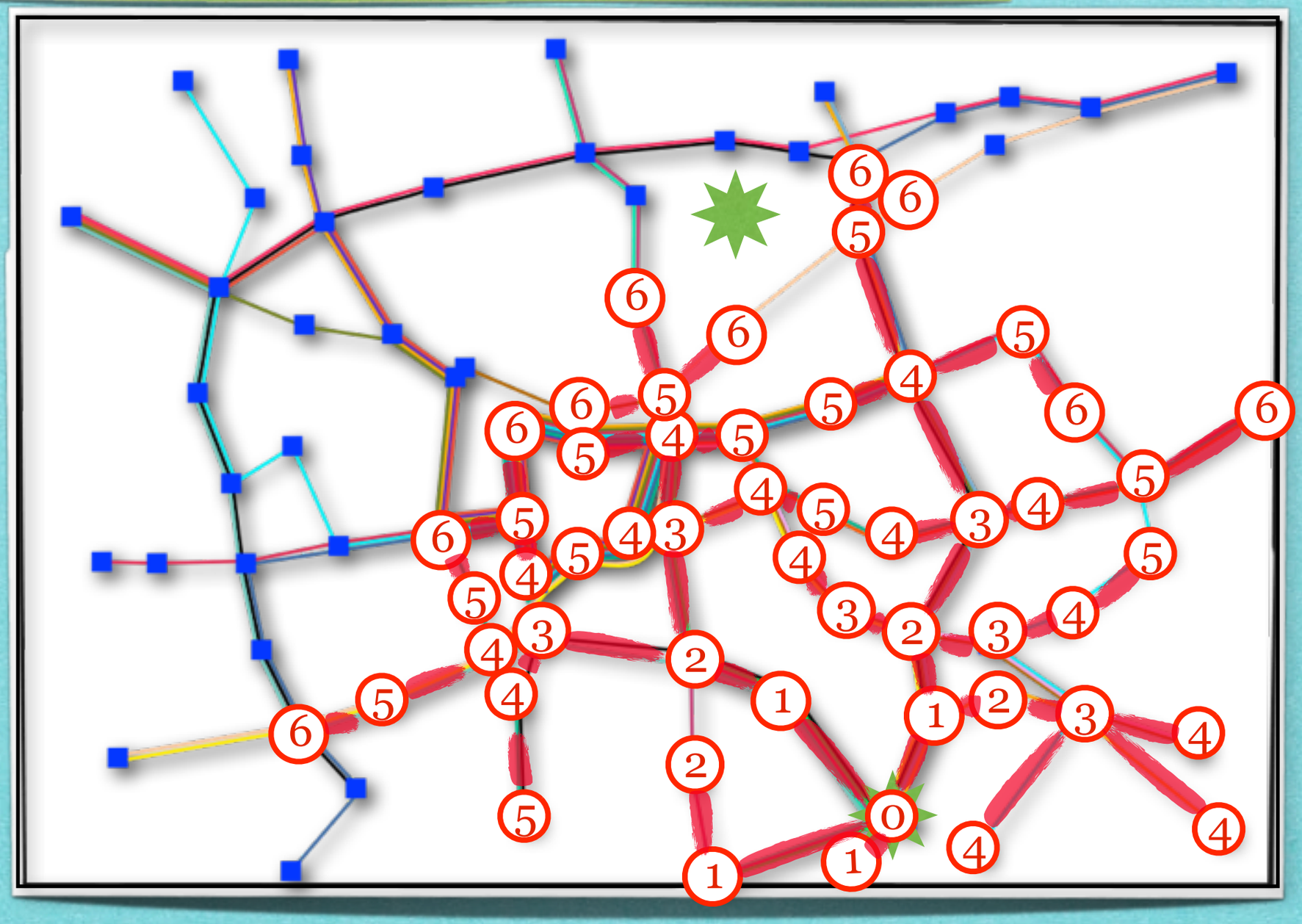
Wellenreiten in Graphen



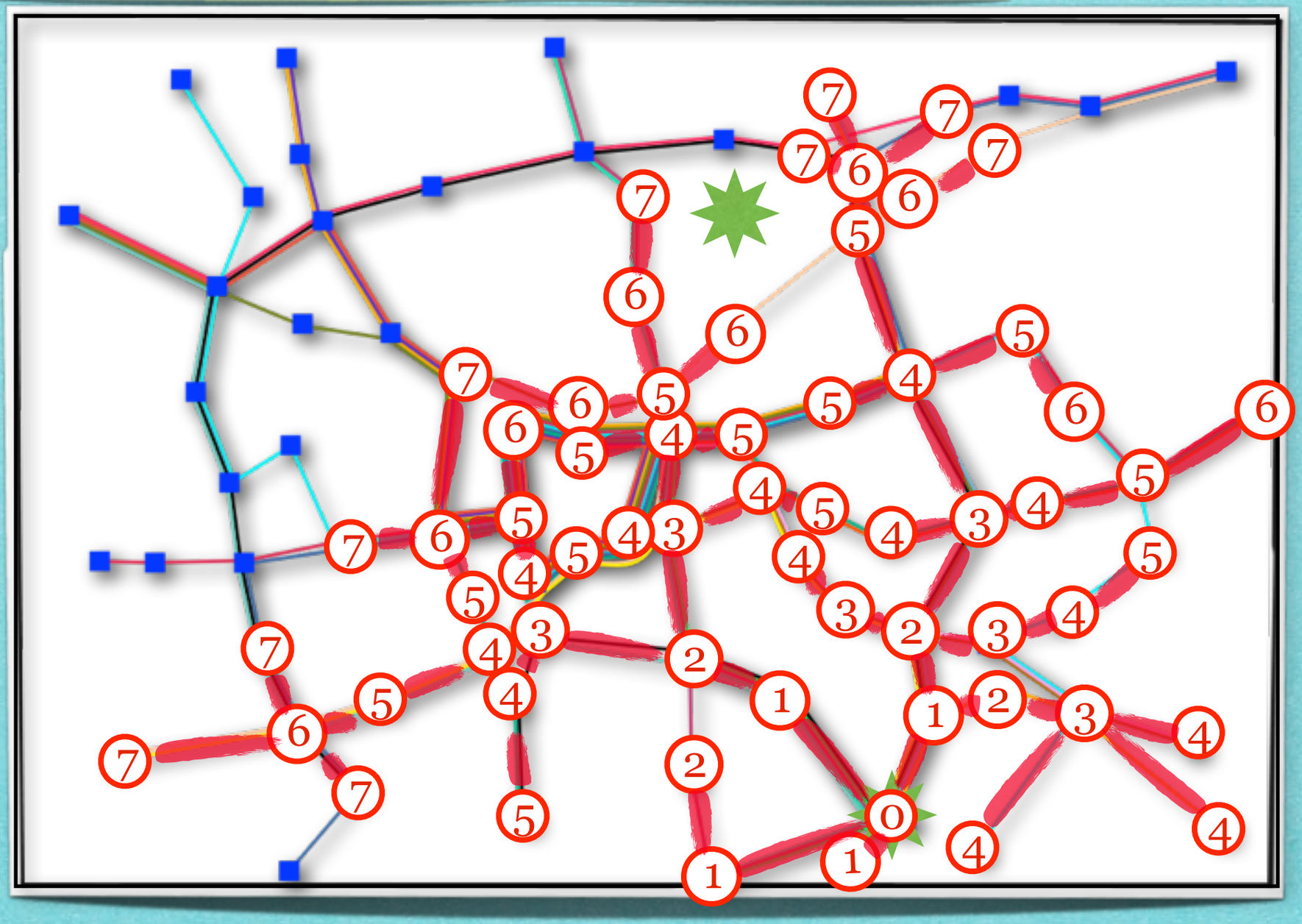
Wellenreiten in Graphen



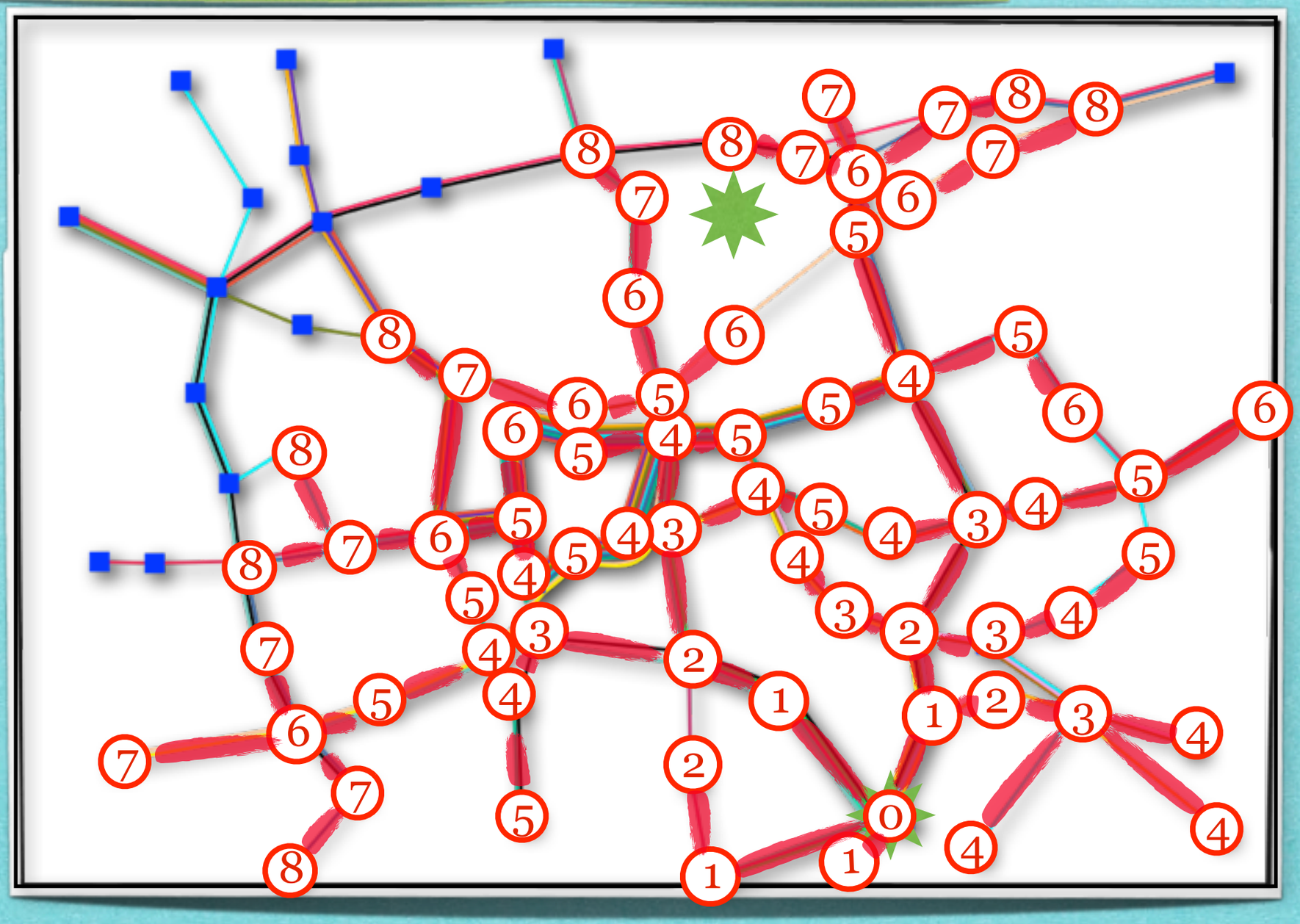
Wellenreiten in Graphen



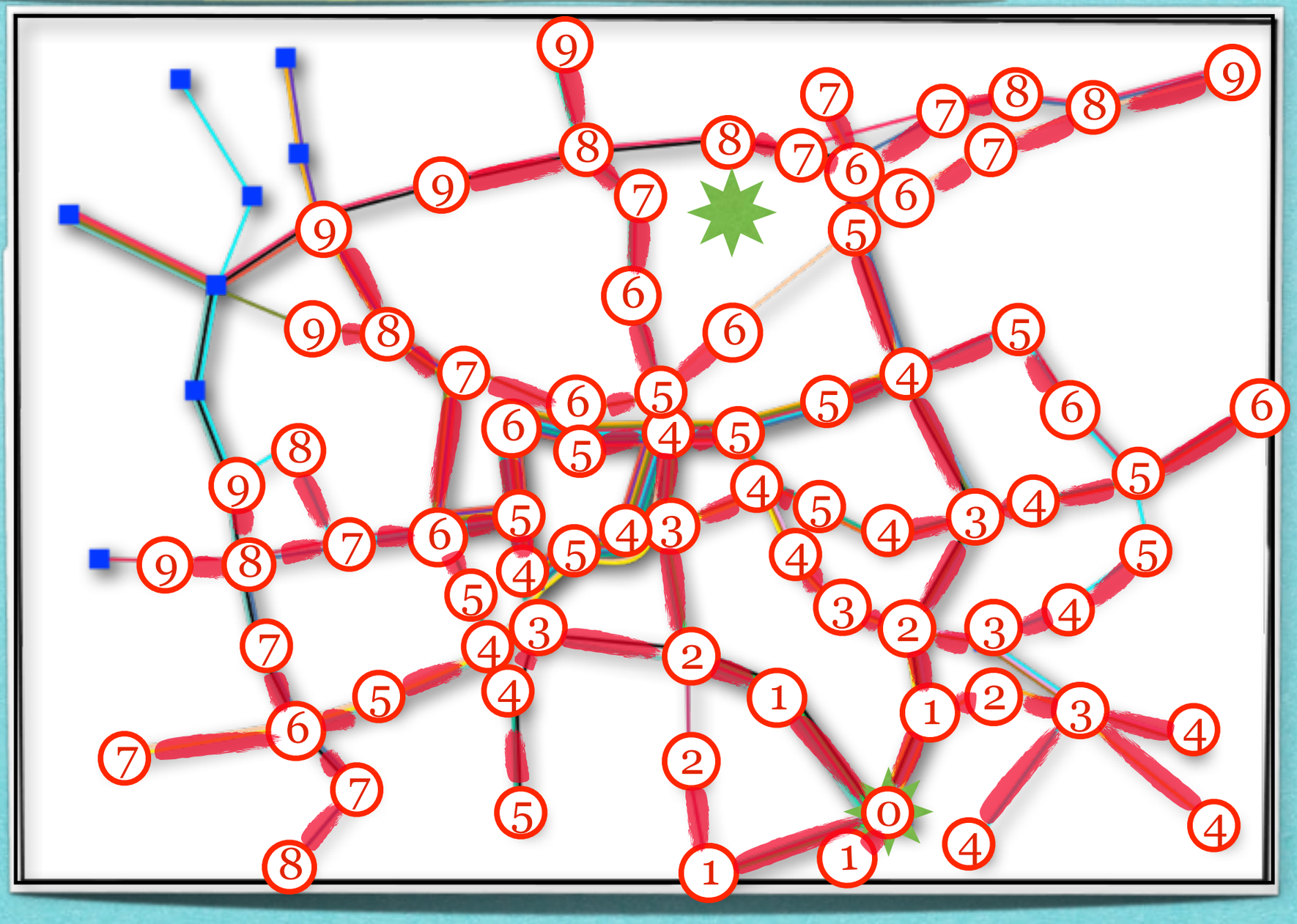
Wellenreiten in Graphen



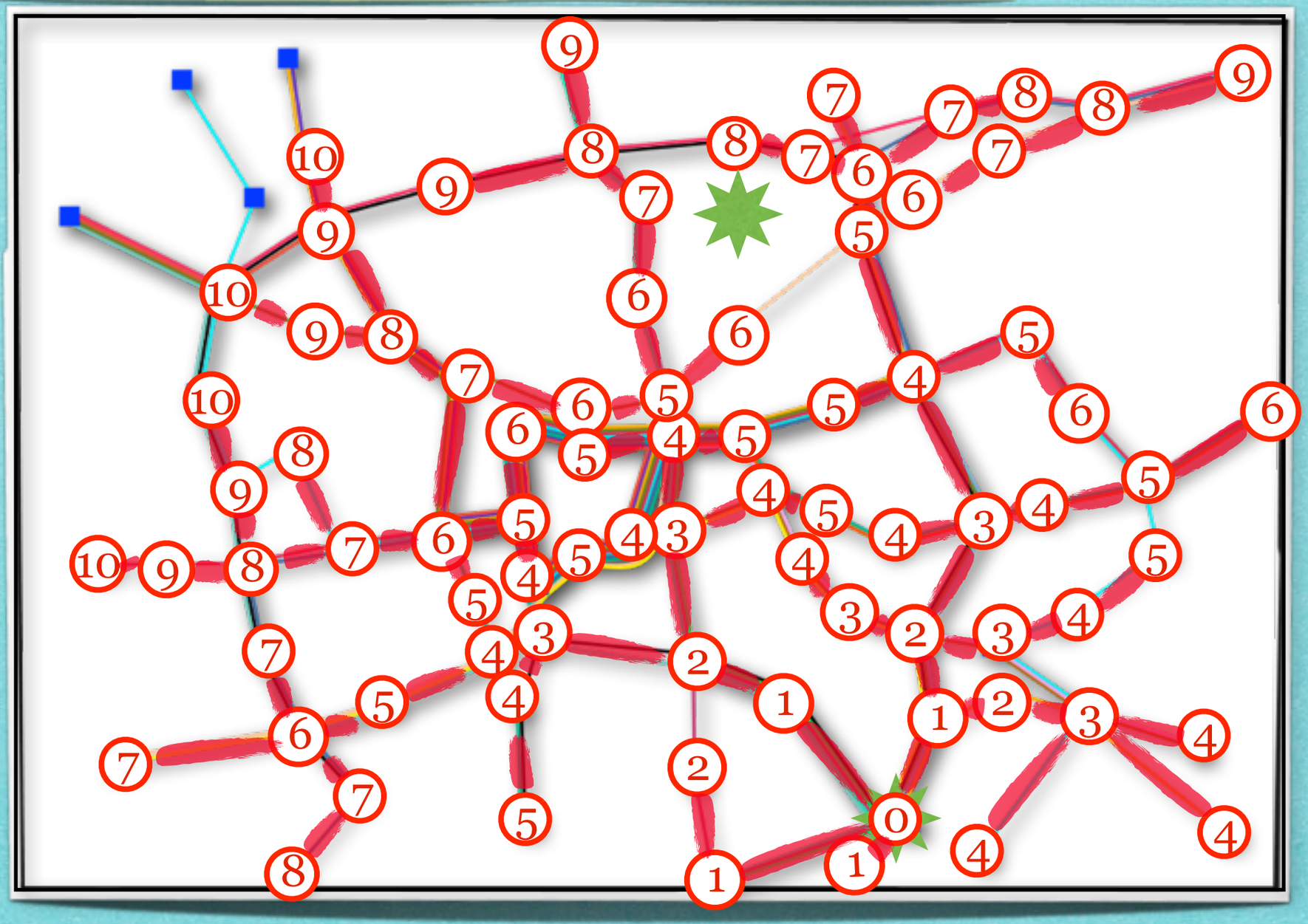
Wellenreiten in Graphen



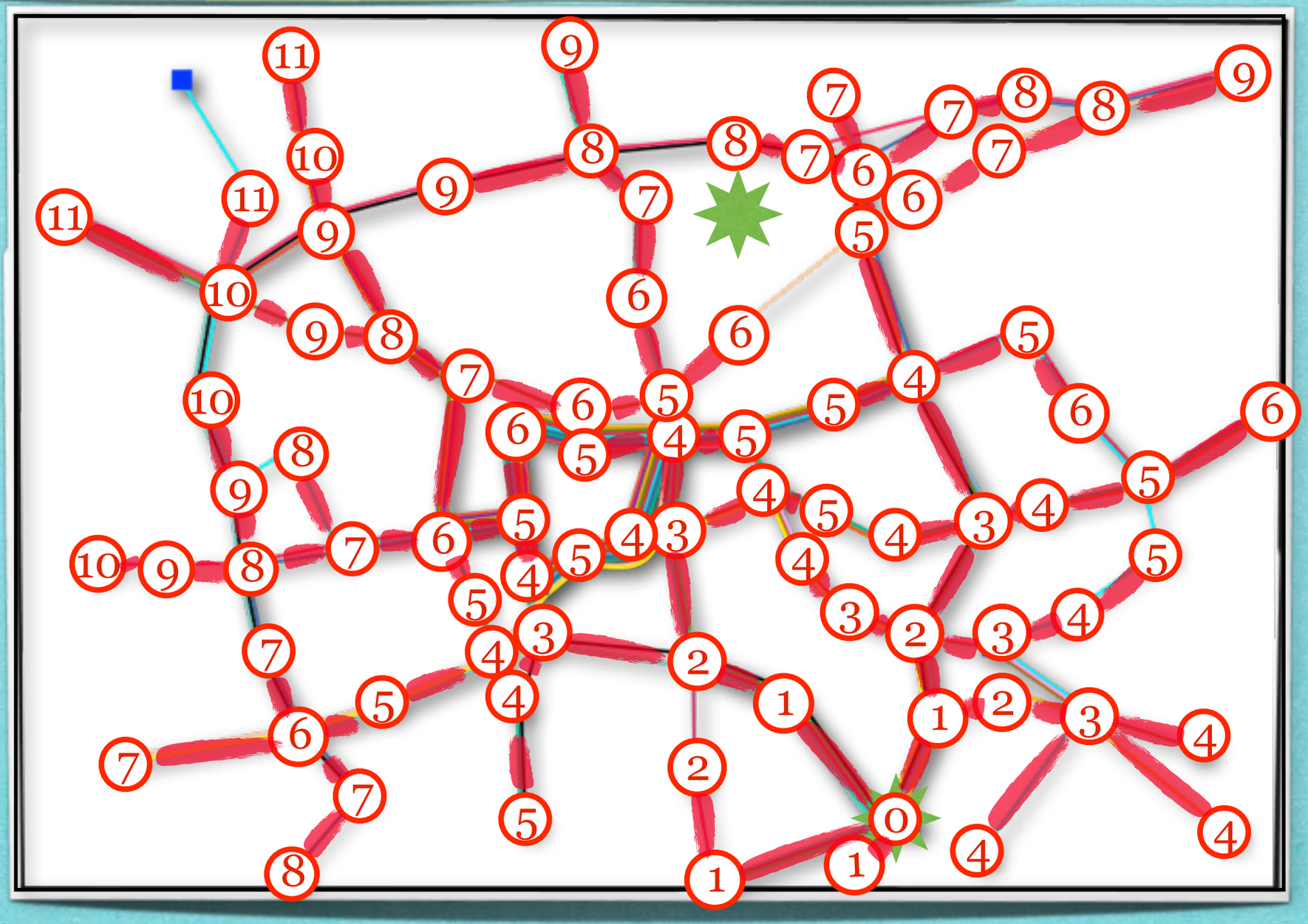
Wellenreiten in Graphen



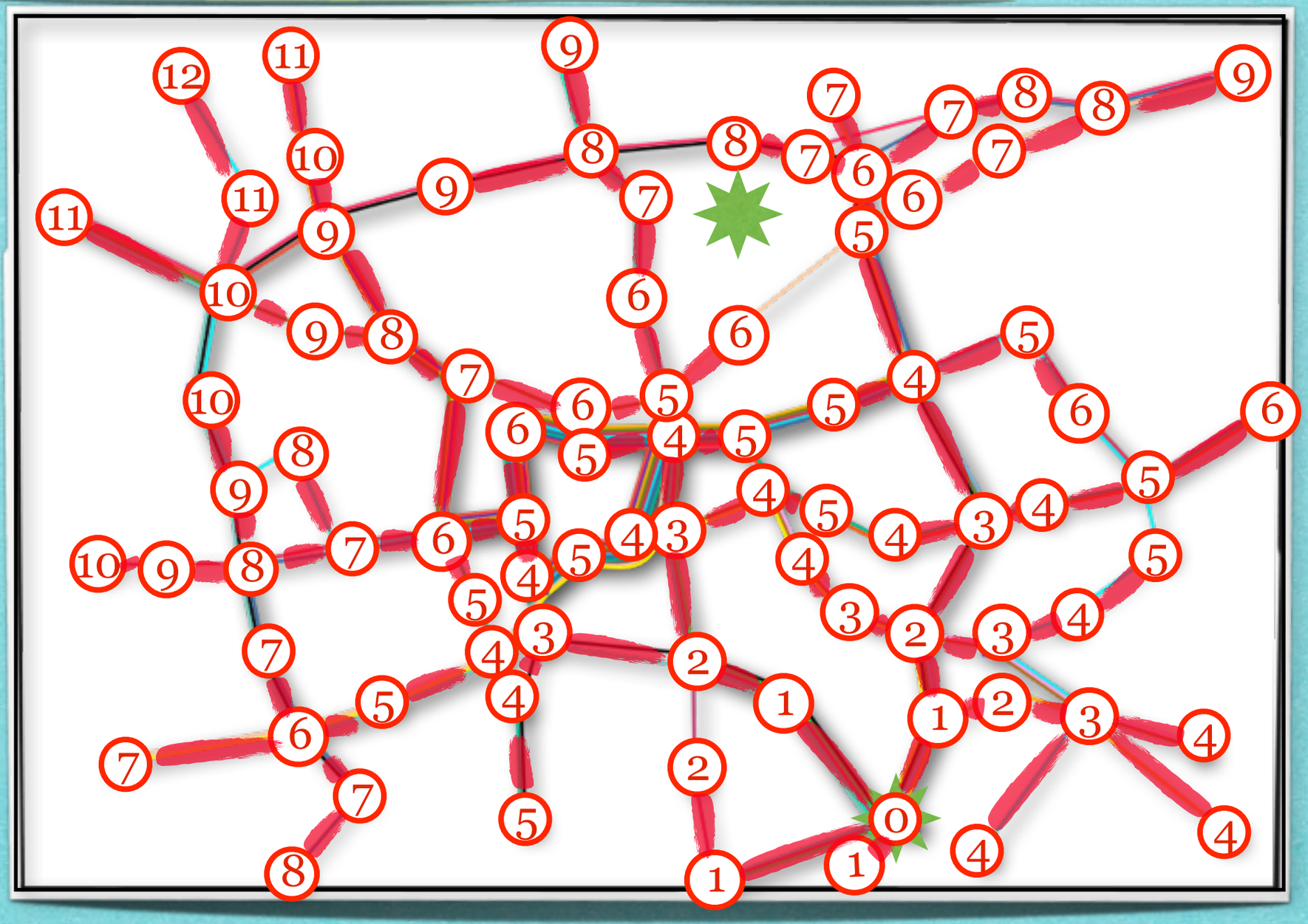
Wellenreiten in Graphen



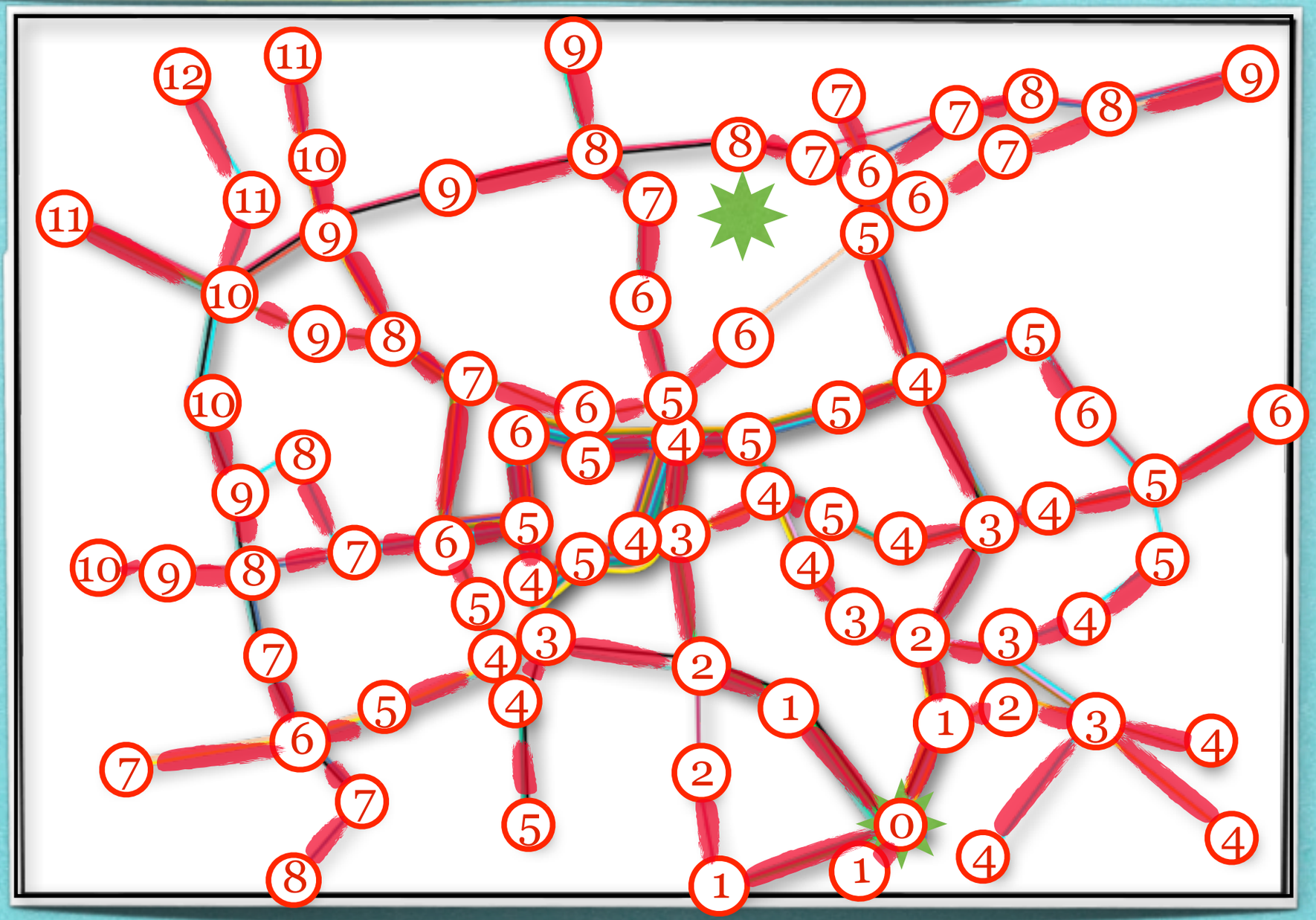
Wellenreiten in Graphen



Wellenreiten in Graphen

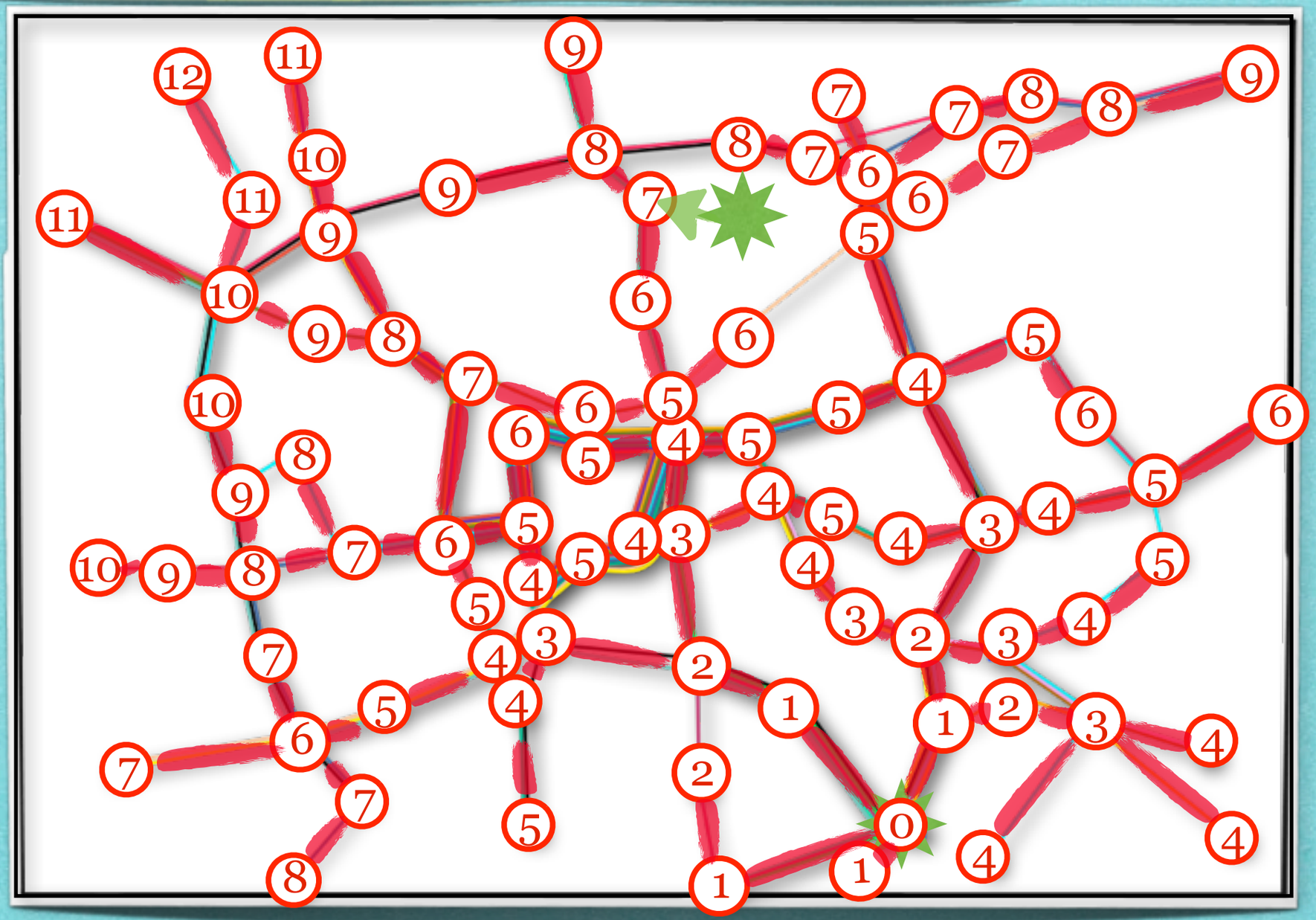


Wellenreiten in Graphen



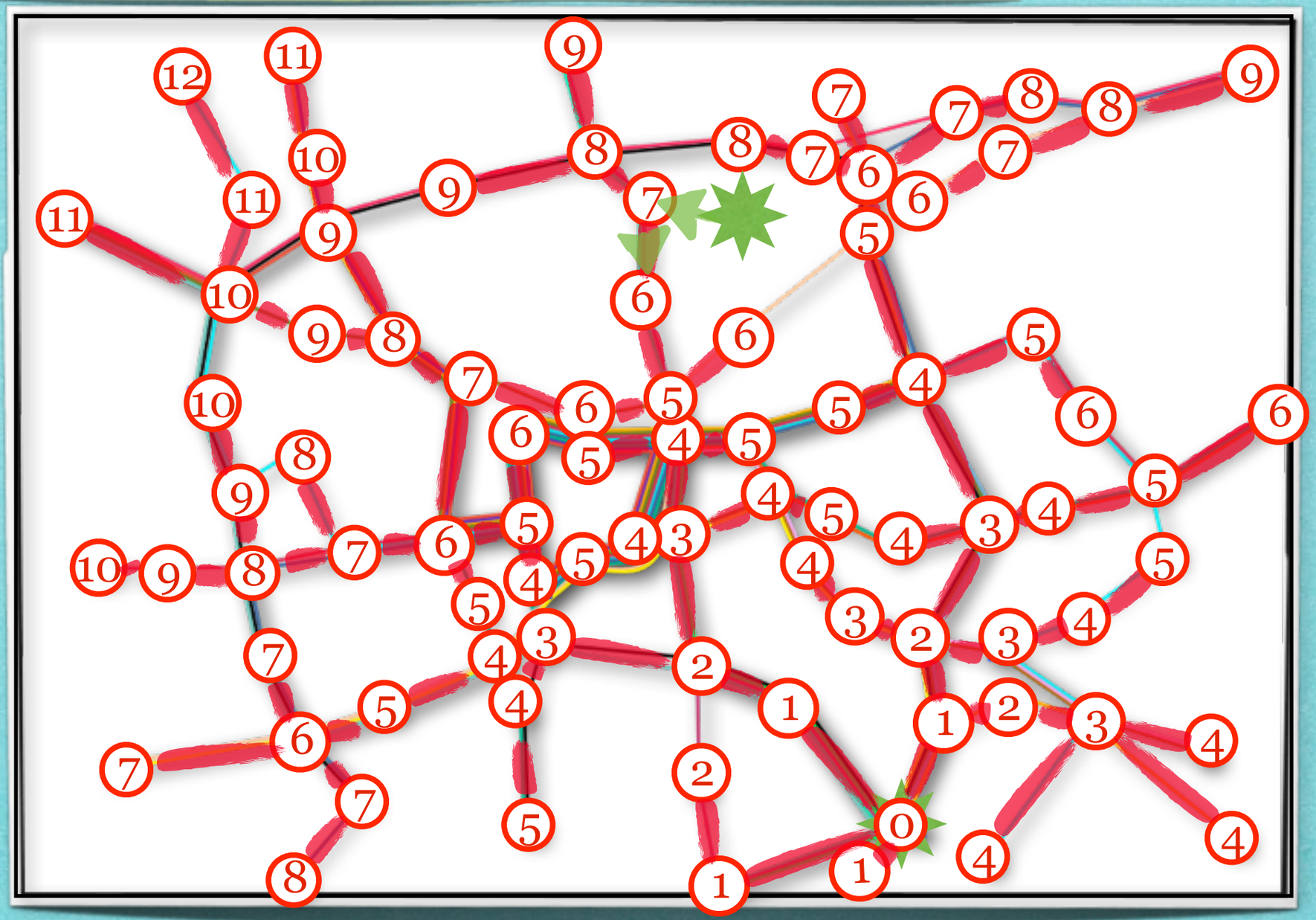
Breitensuche

Wellenreiten in Graphen



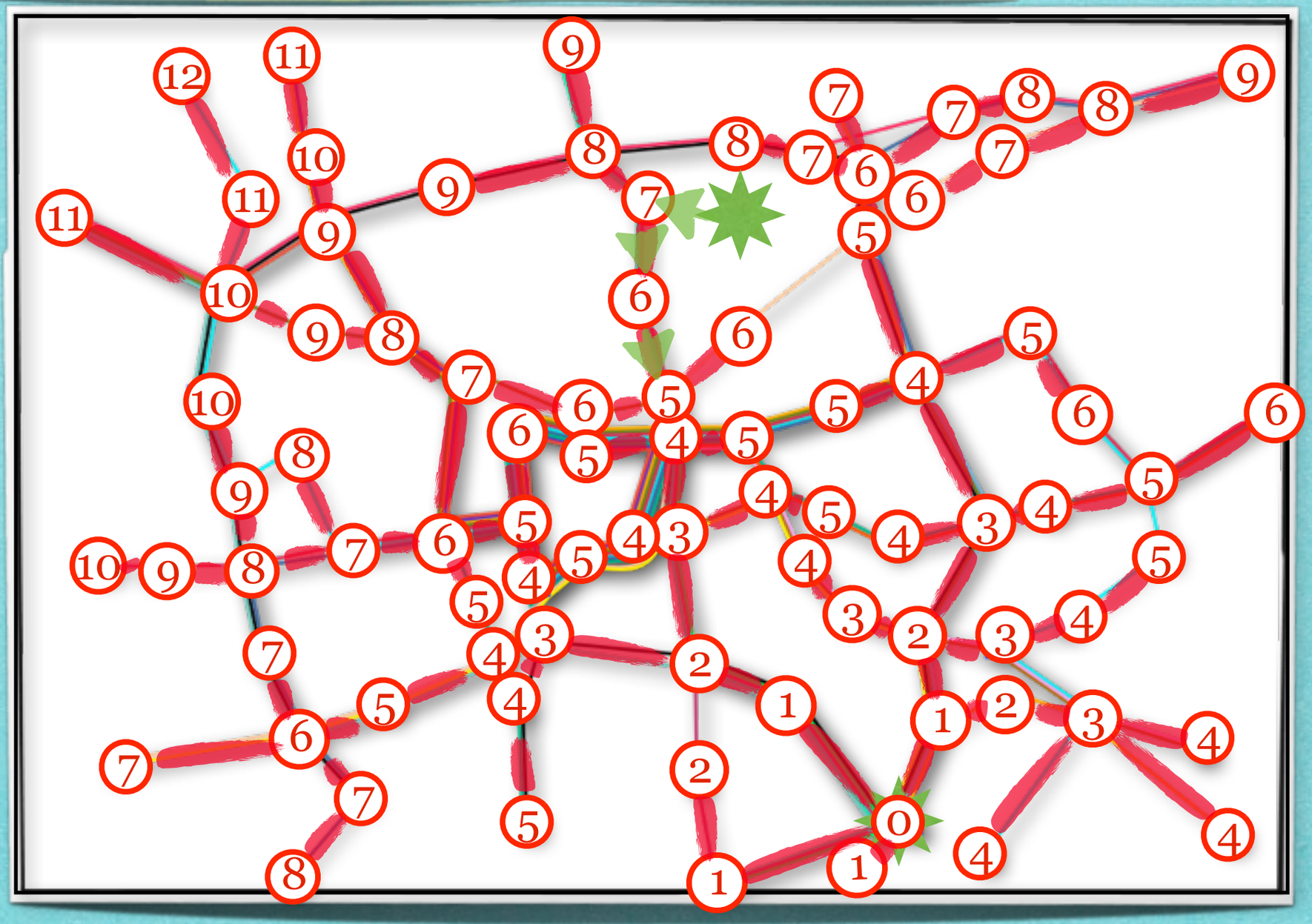
Breitensuche

Wellenreiten in Graphen



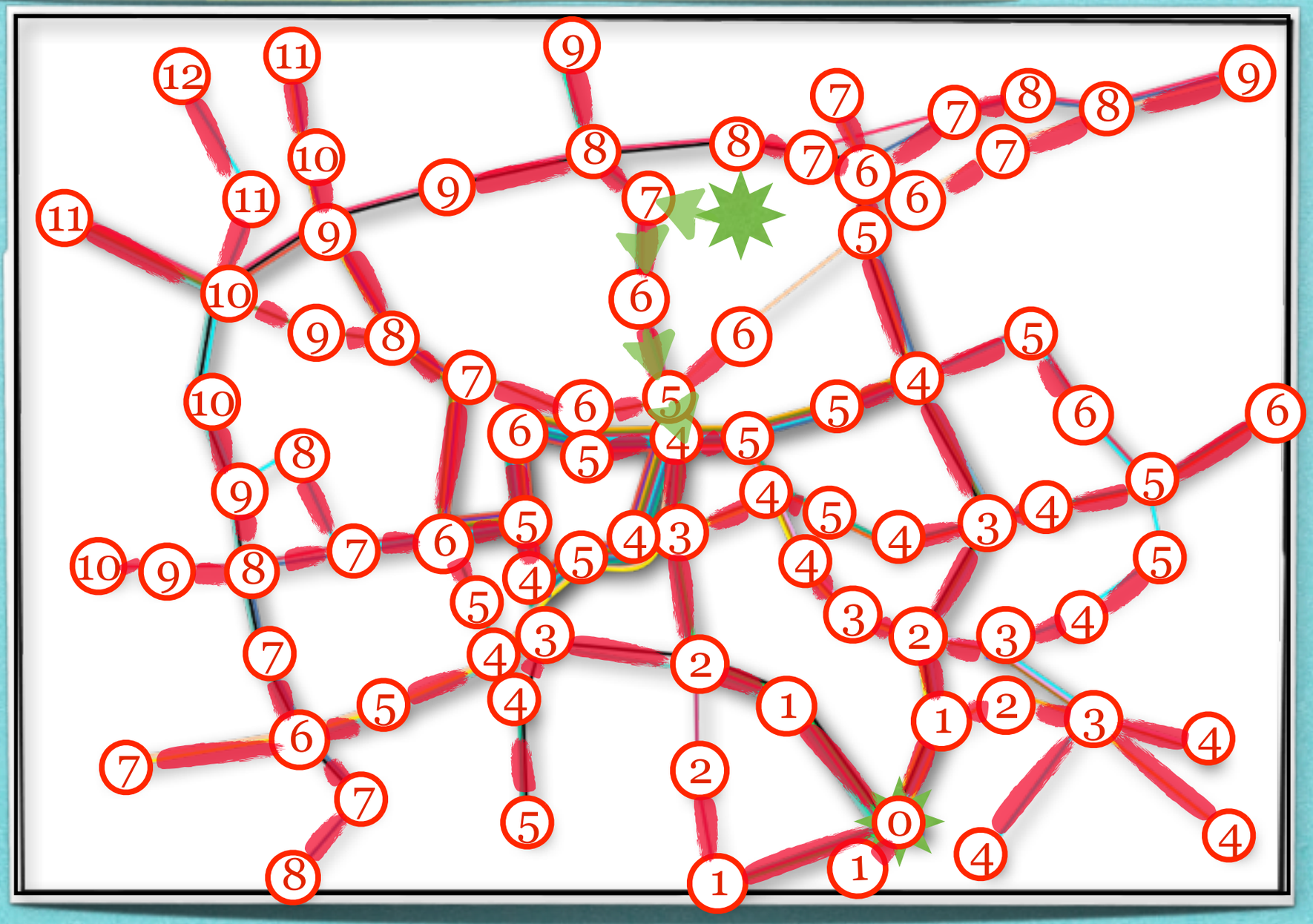
Breitensuche

Wellenreiten in Graphen



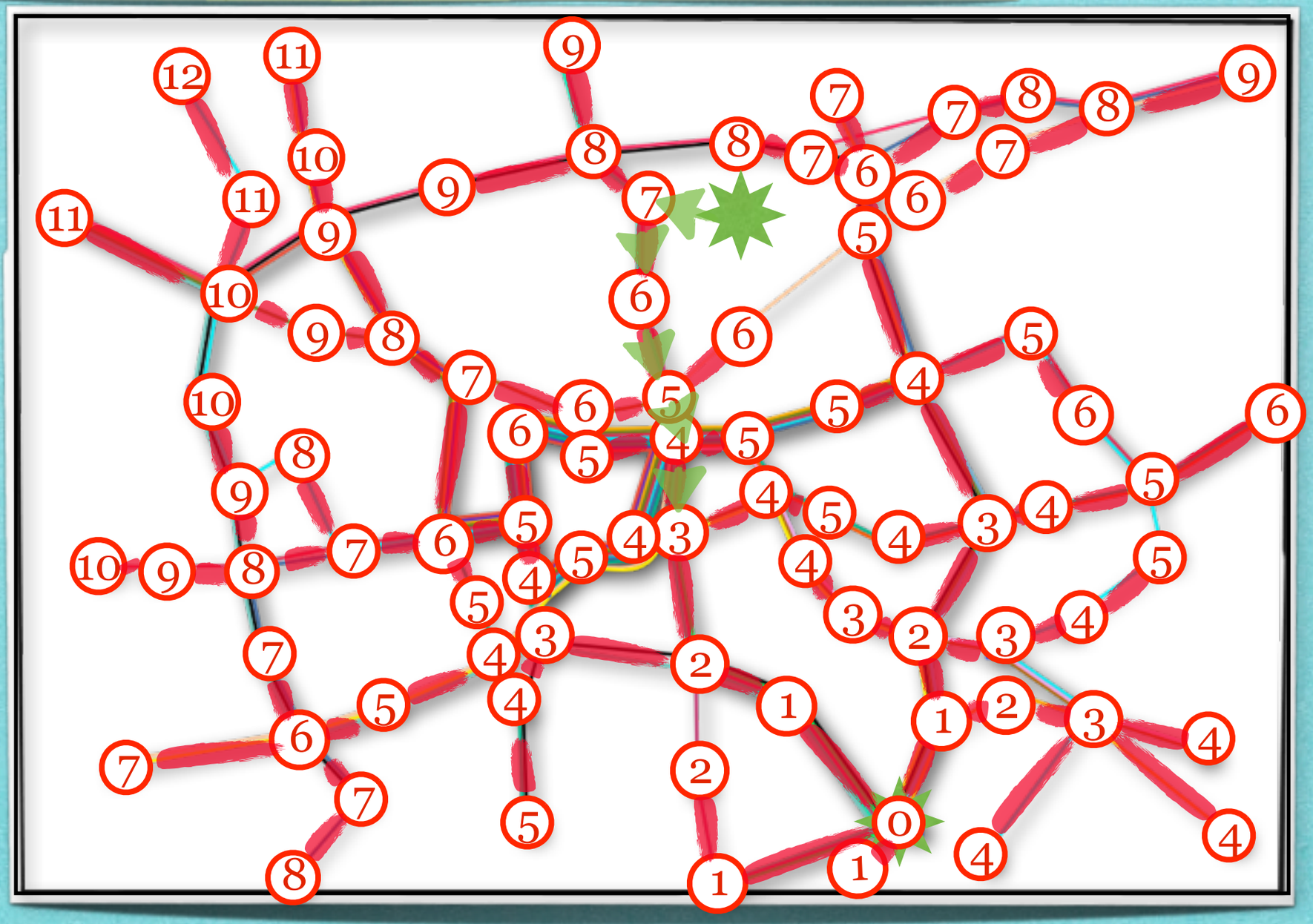
Breitensuche

Wellenreiten in Graphen



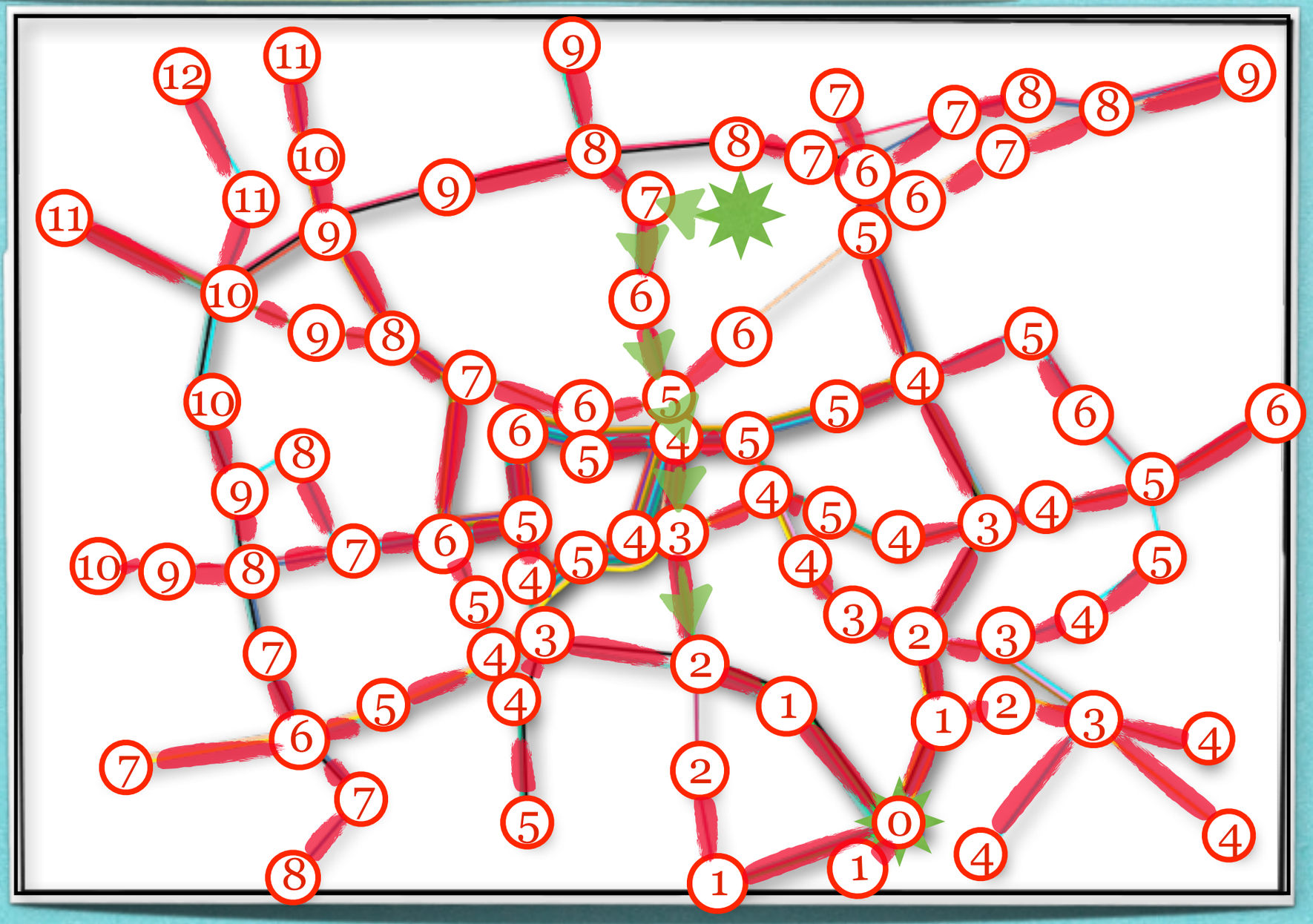
Breitensuche

Wellenreiten in Graphen



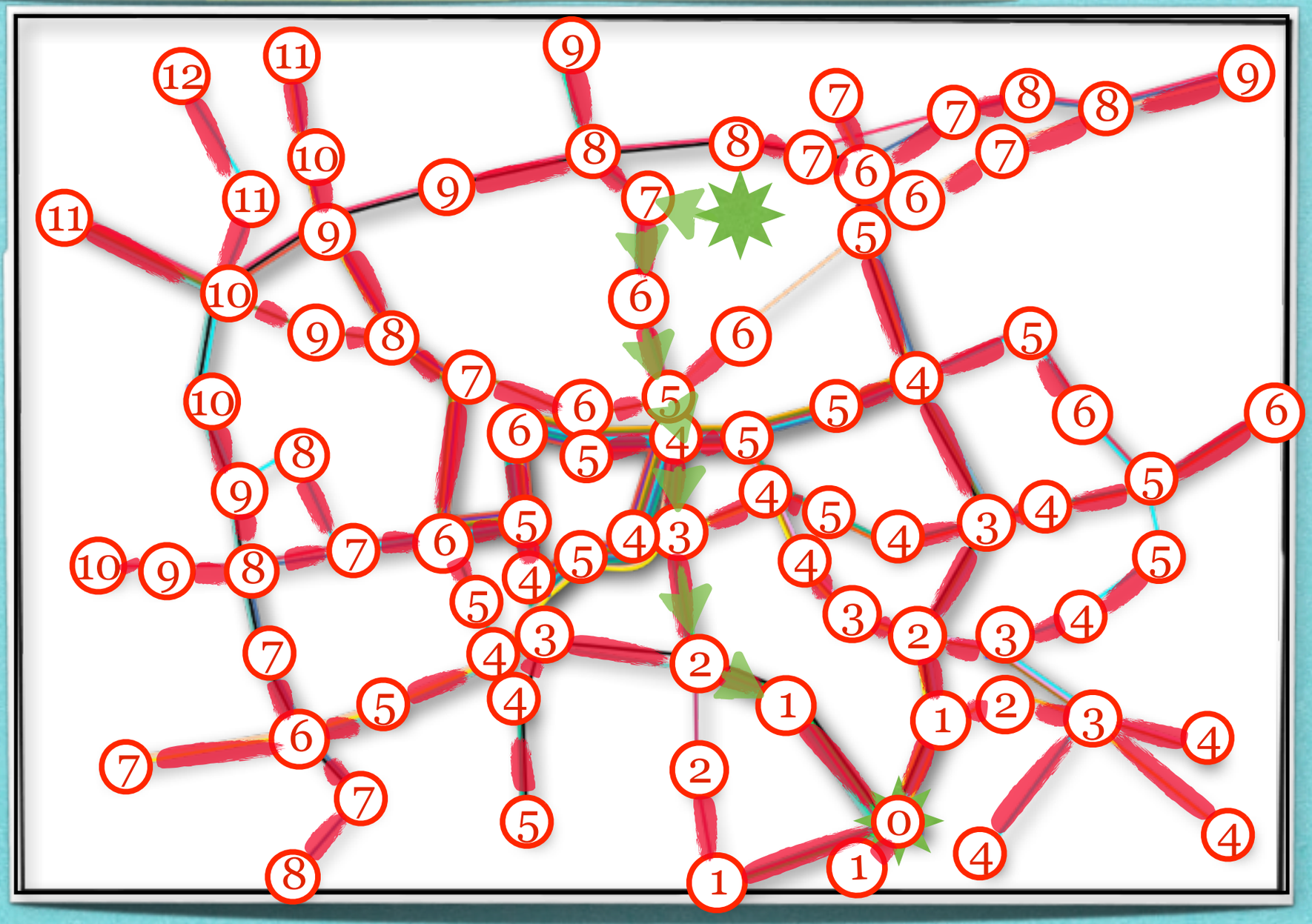
Breitensuche

Wellenreiten in Graphen



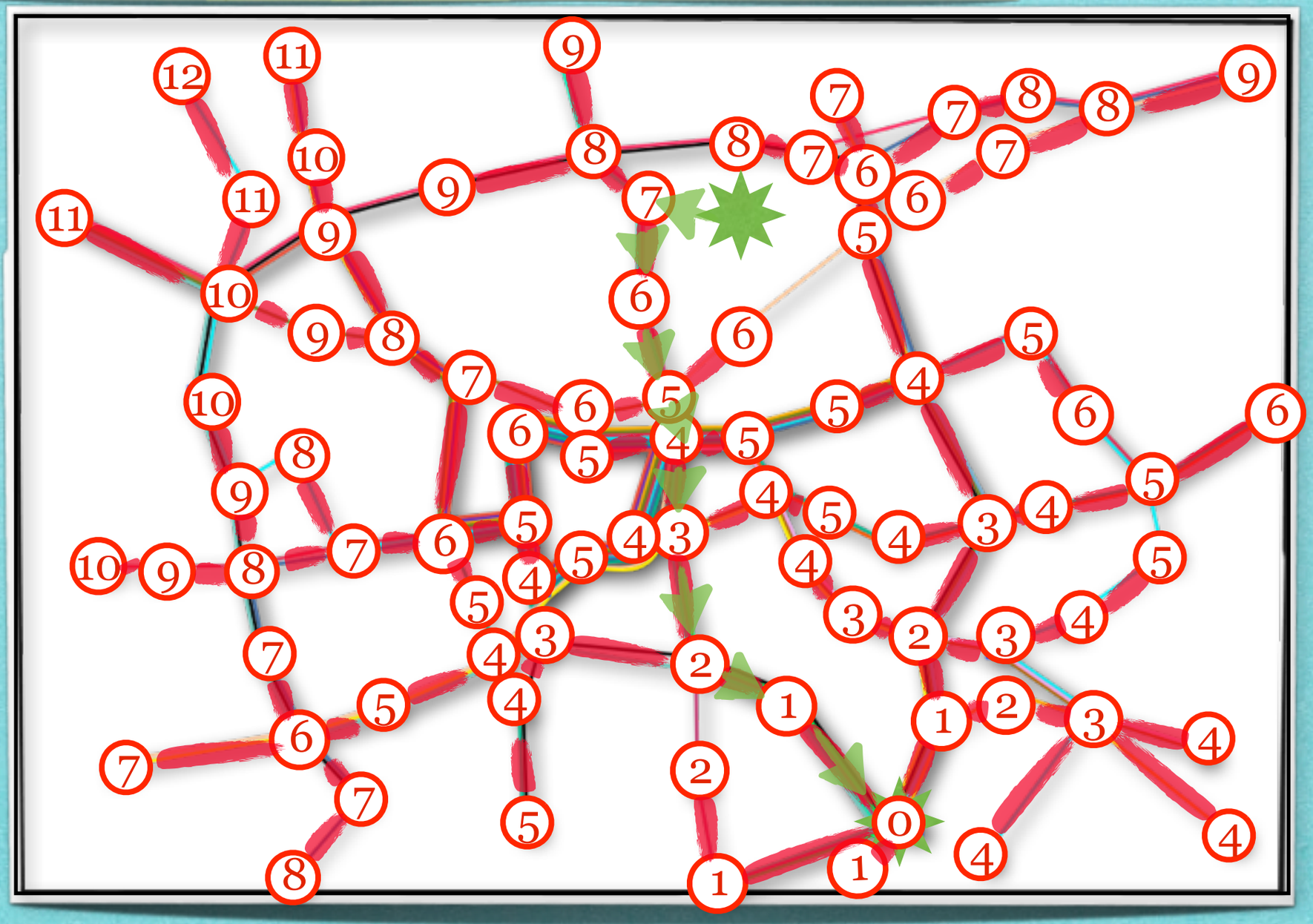
Breitensuche

Wellenreiten in Graphen



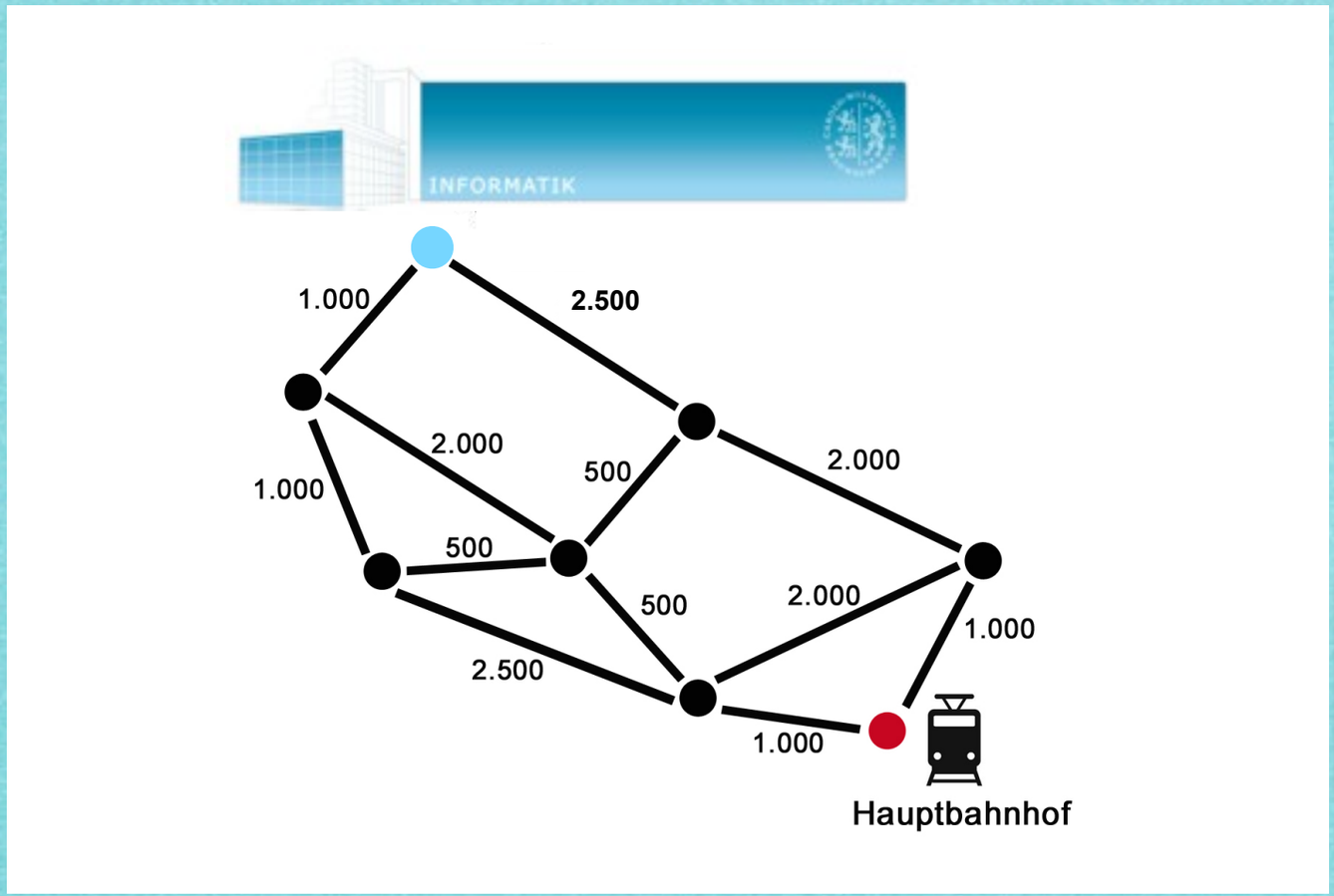
Breitensuche

Wellenreiten in Graphen



Breitensuche

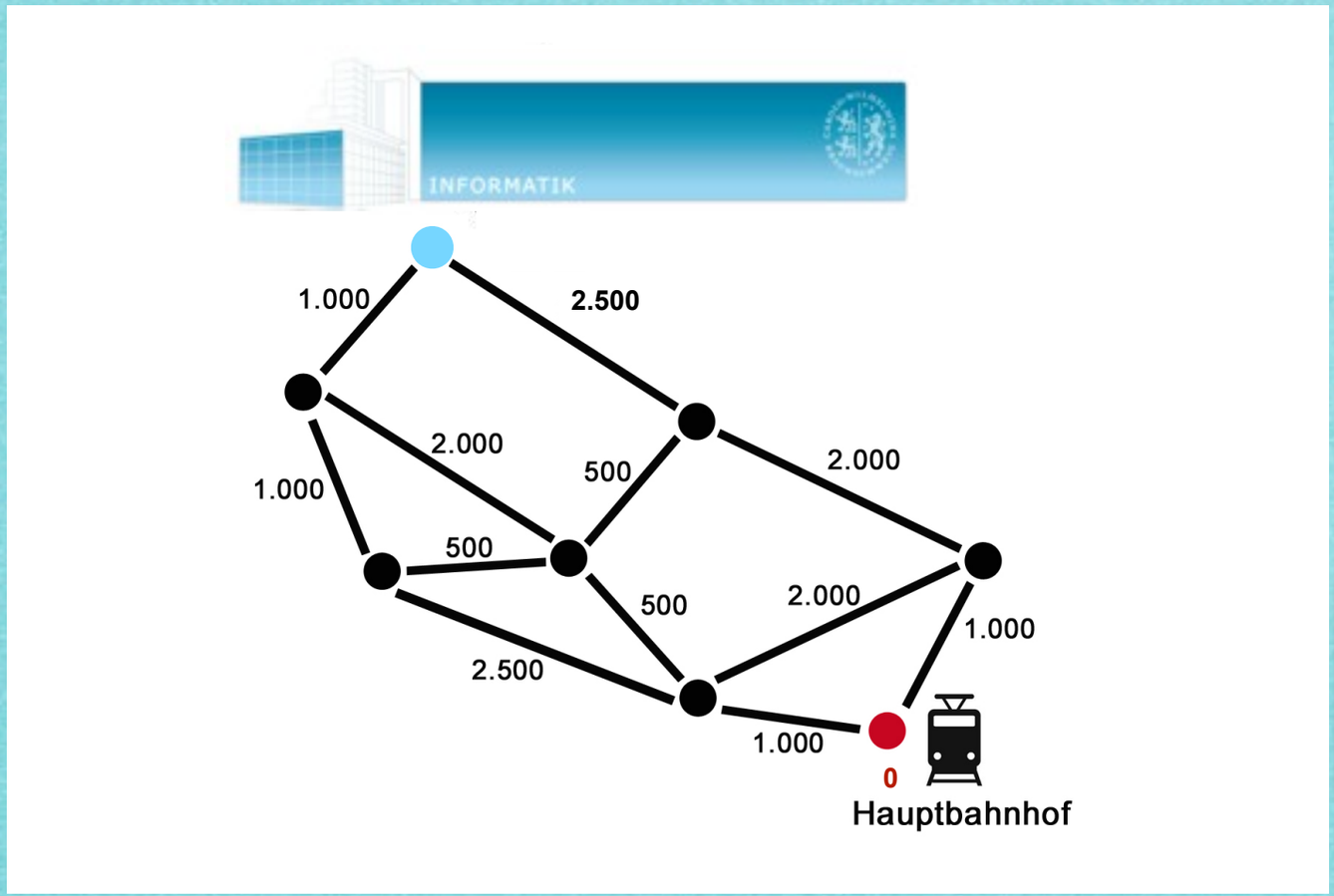
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch
für verschieden
lange Kanten

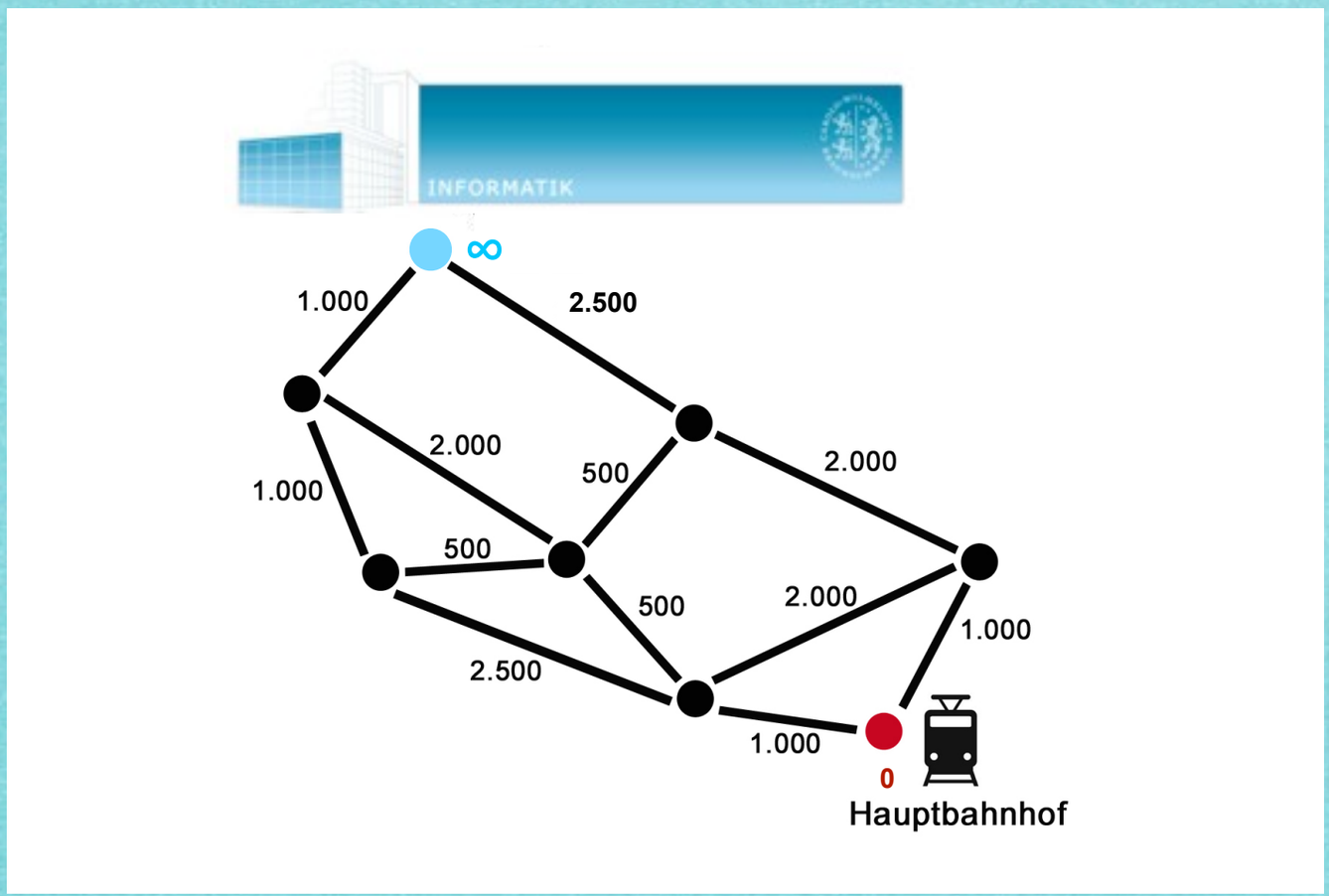
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

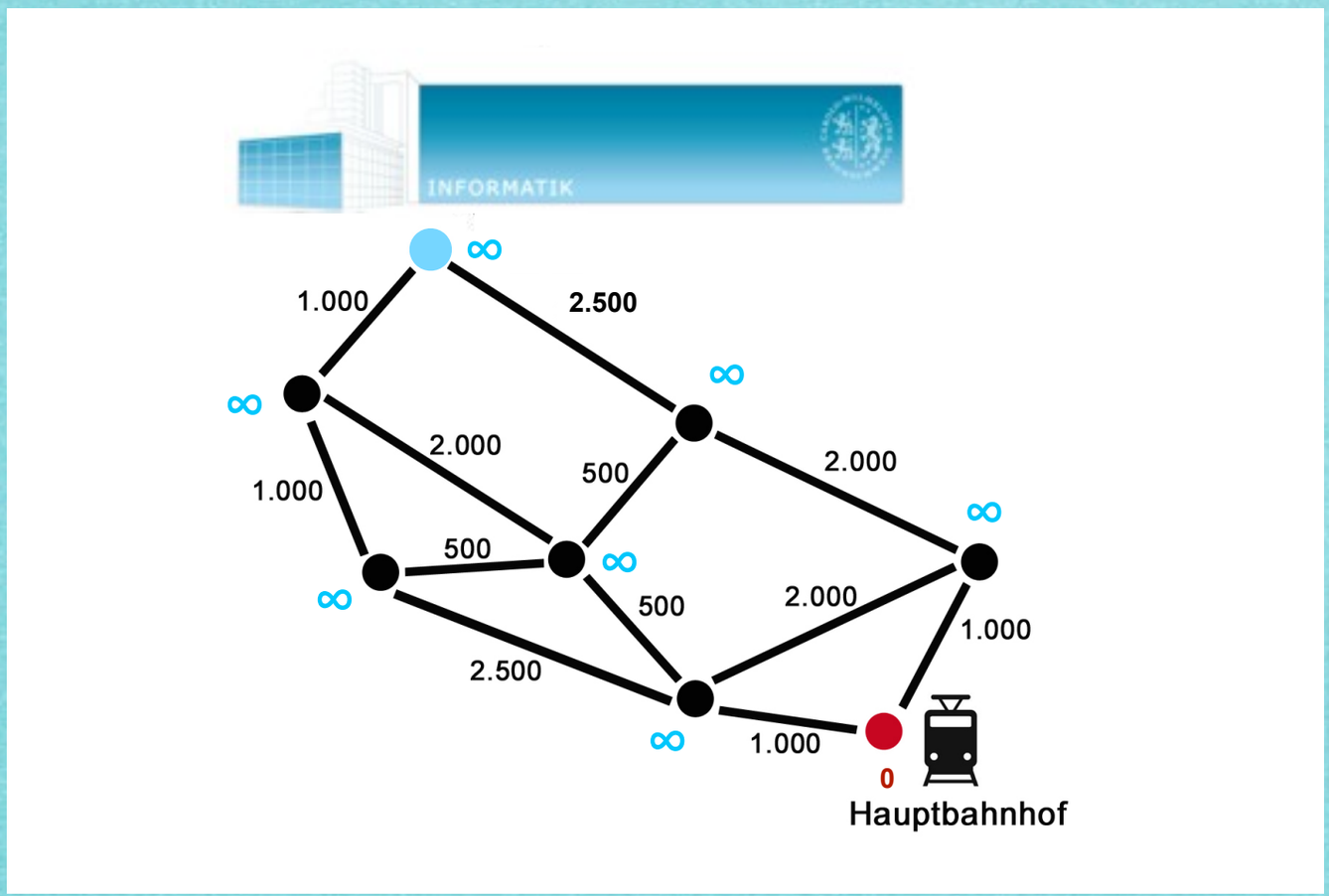
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

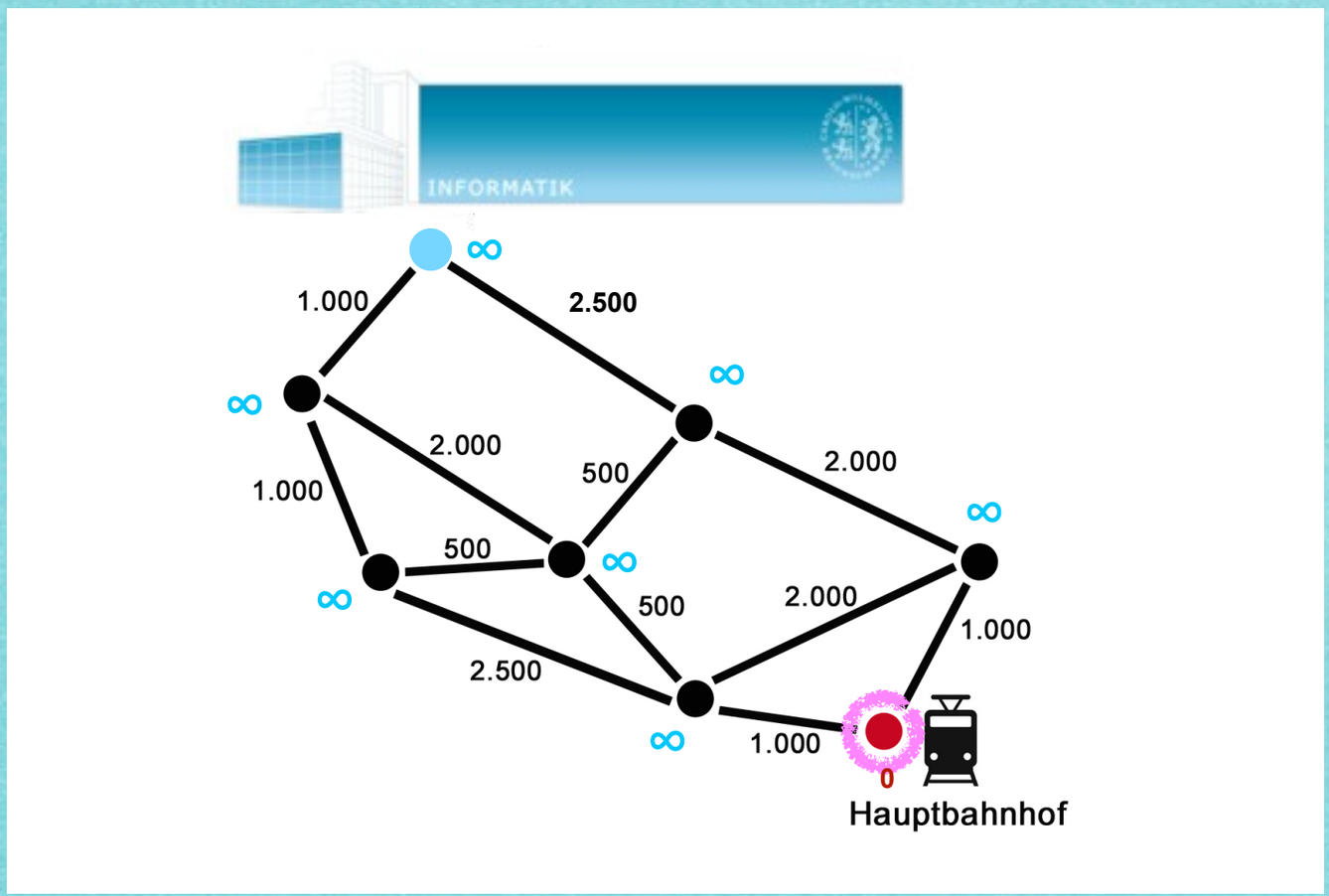
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

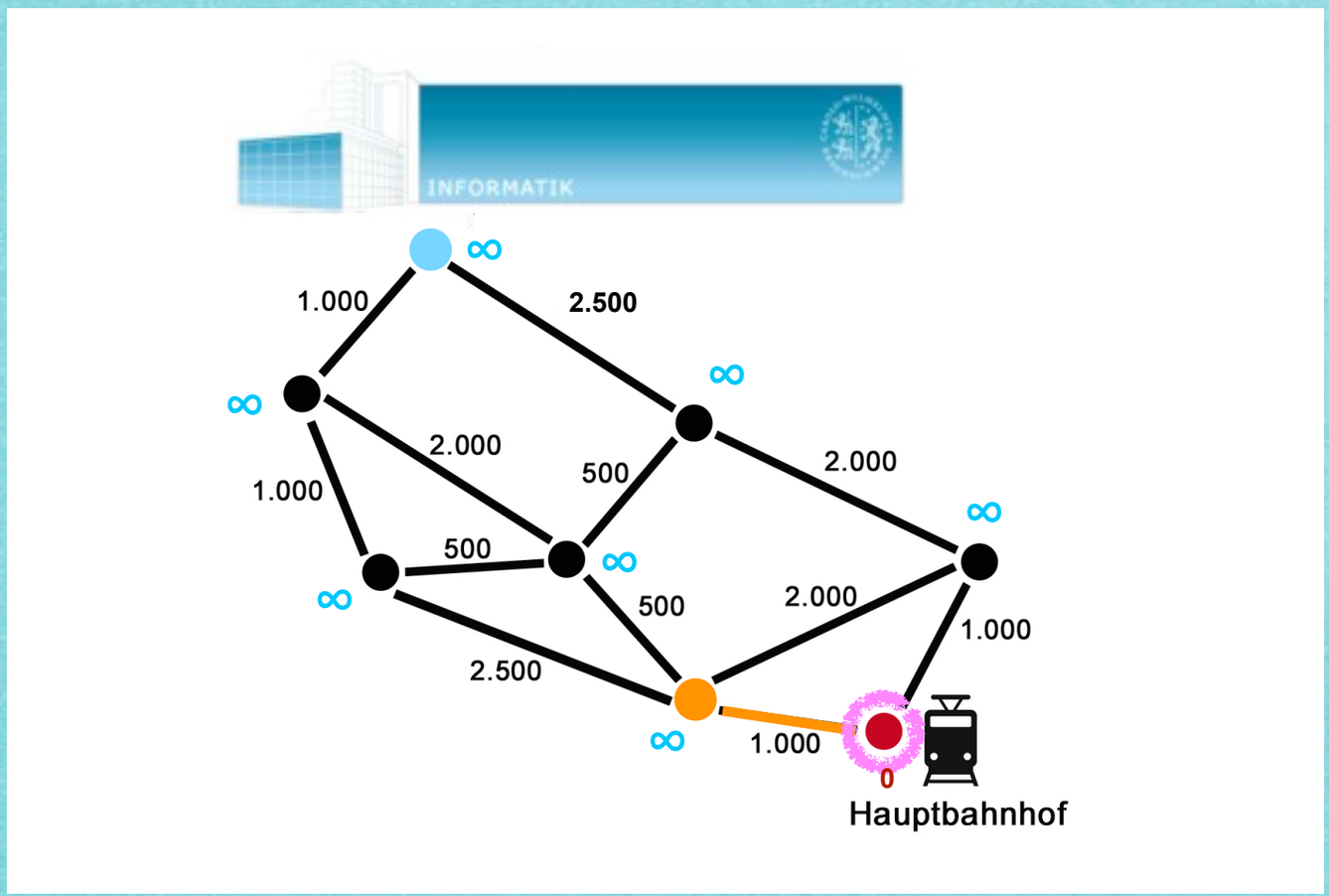
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

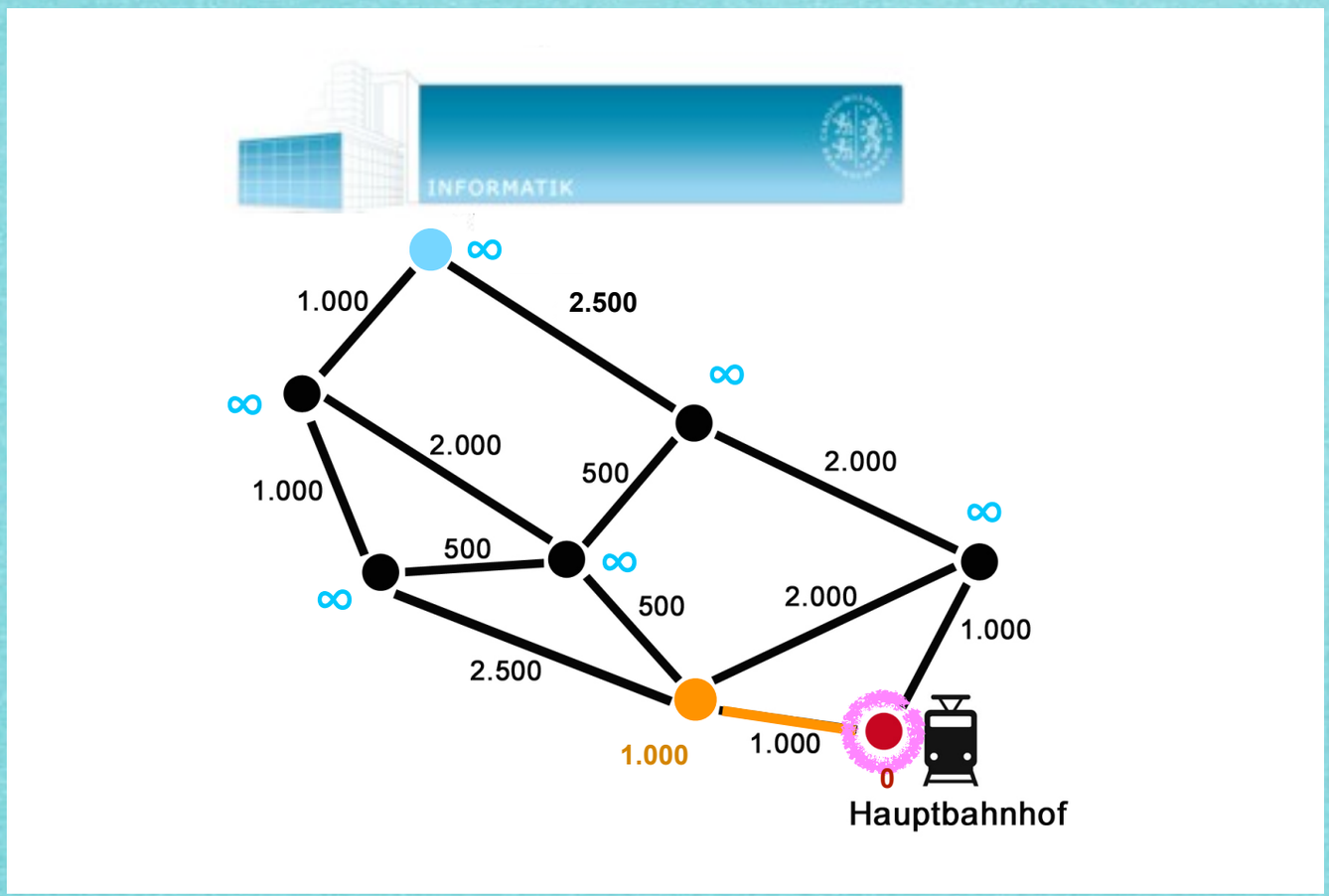
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

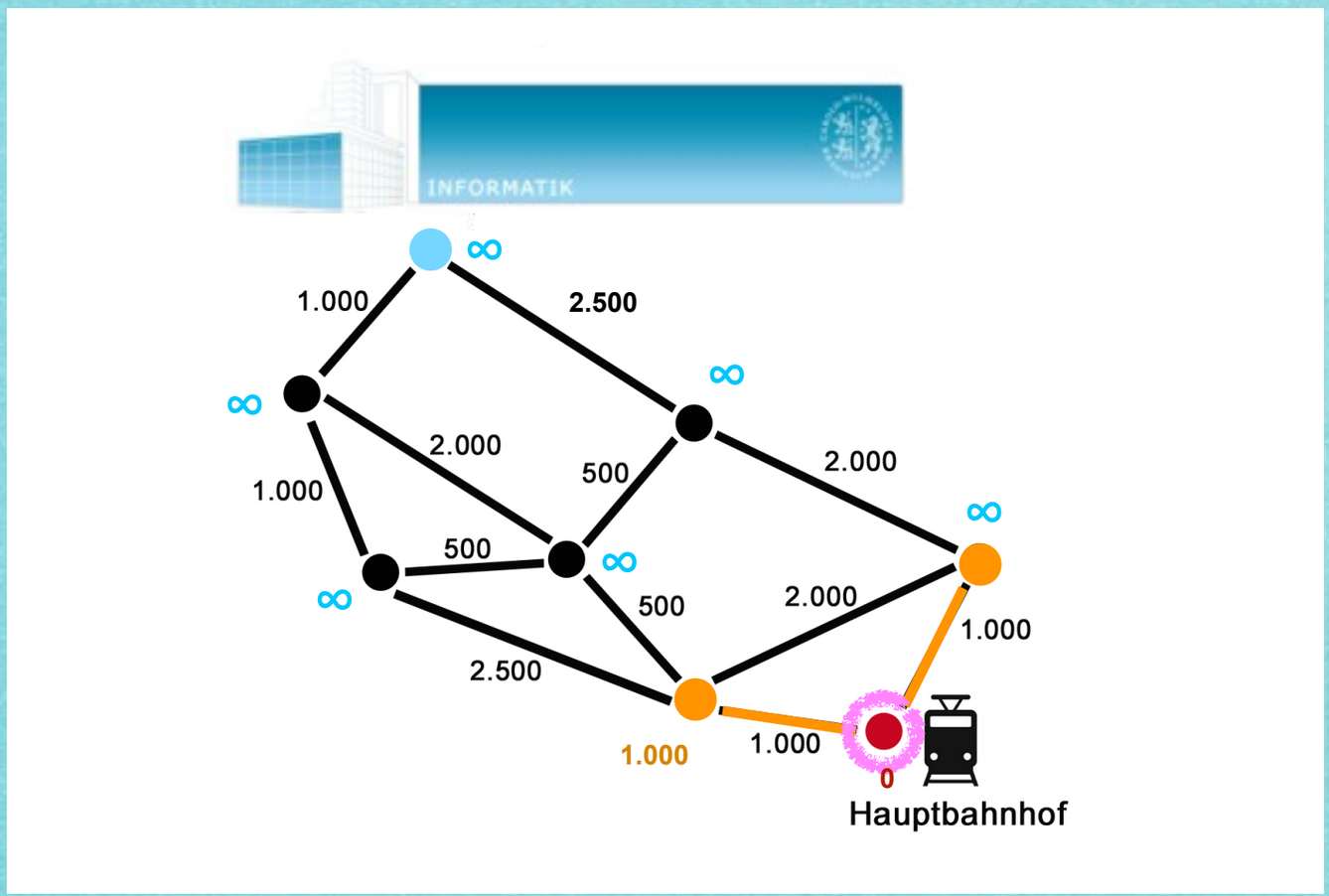
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

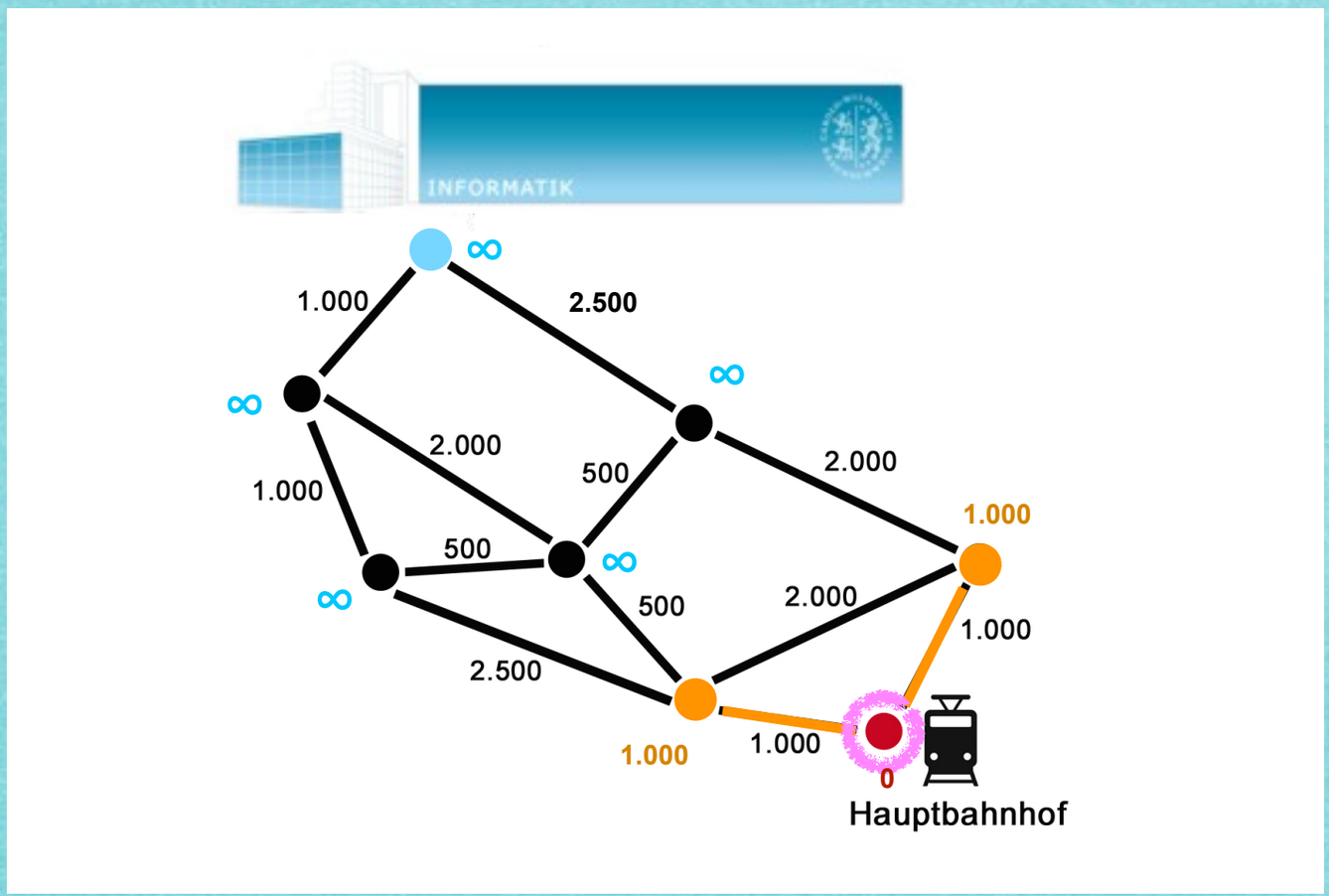
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

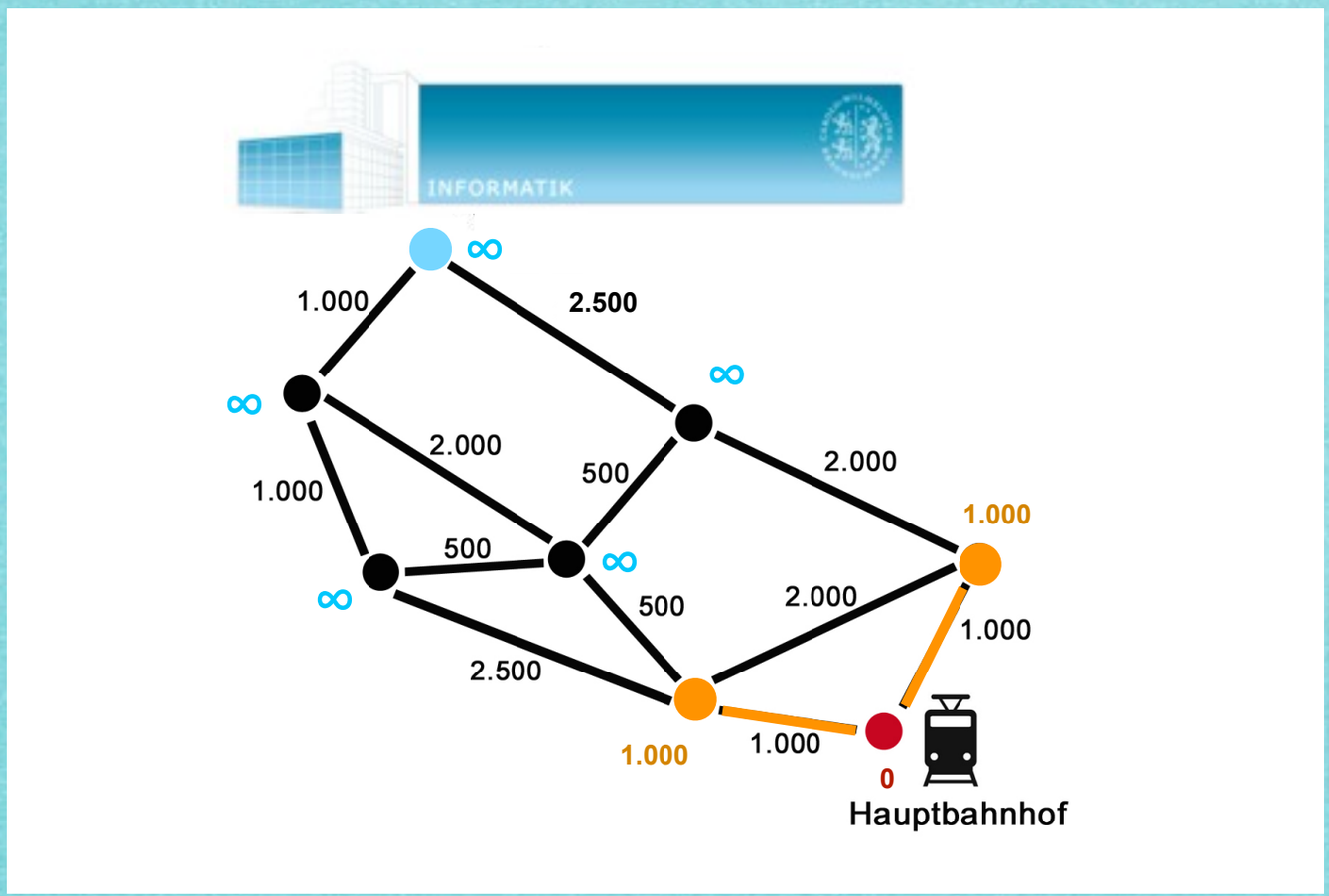
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

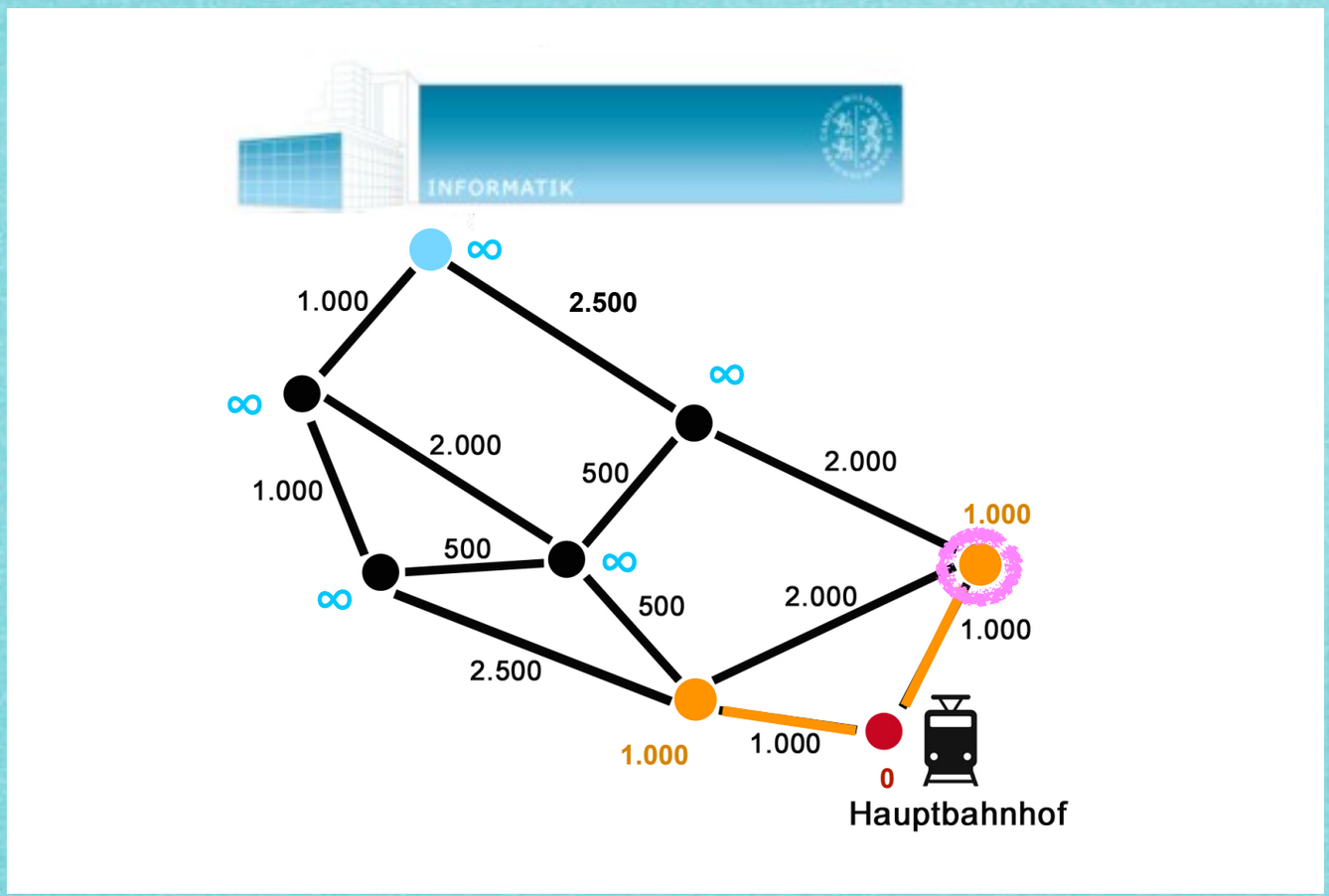
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

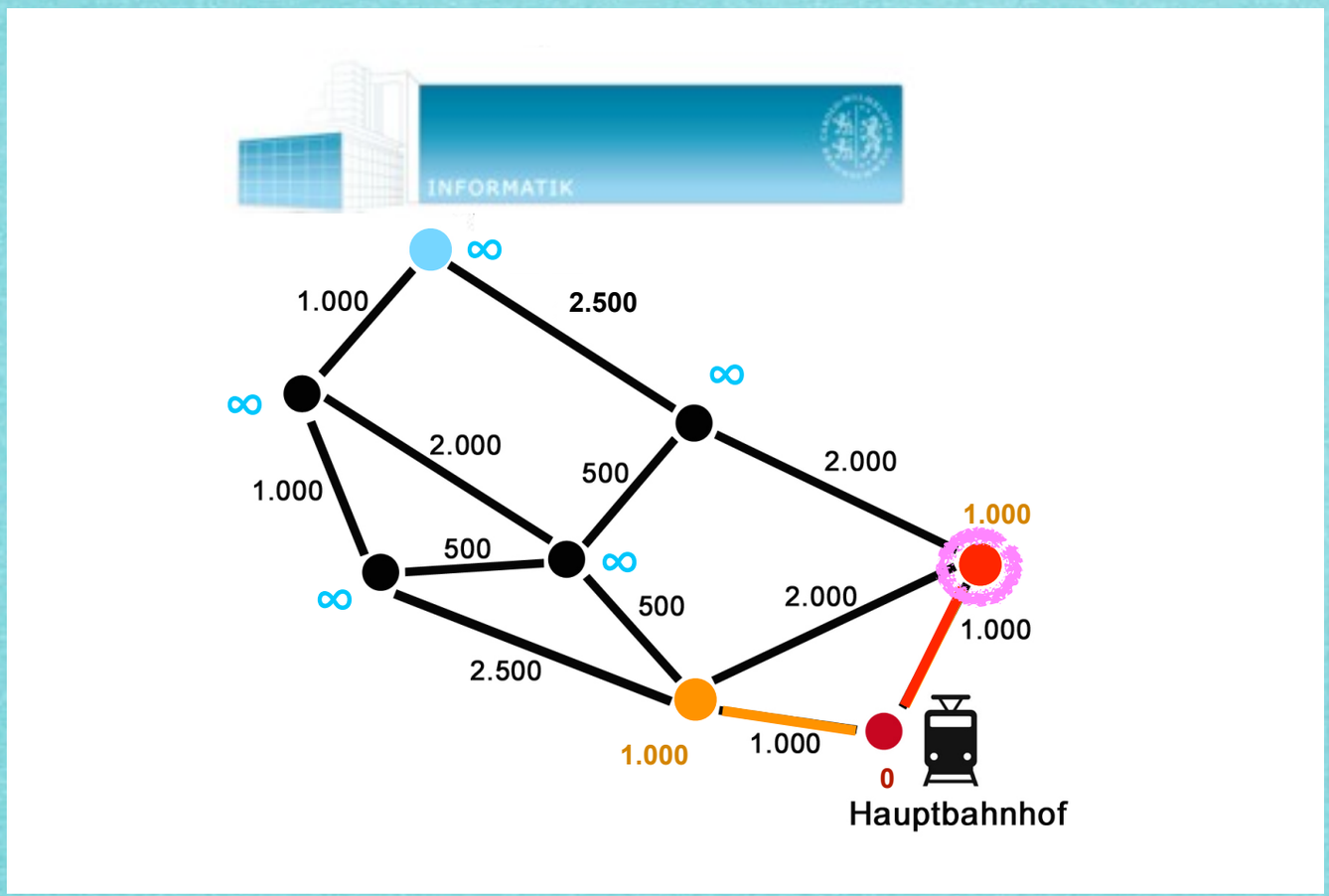
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

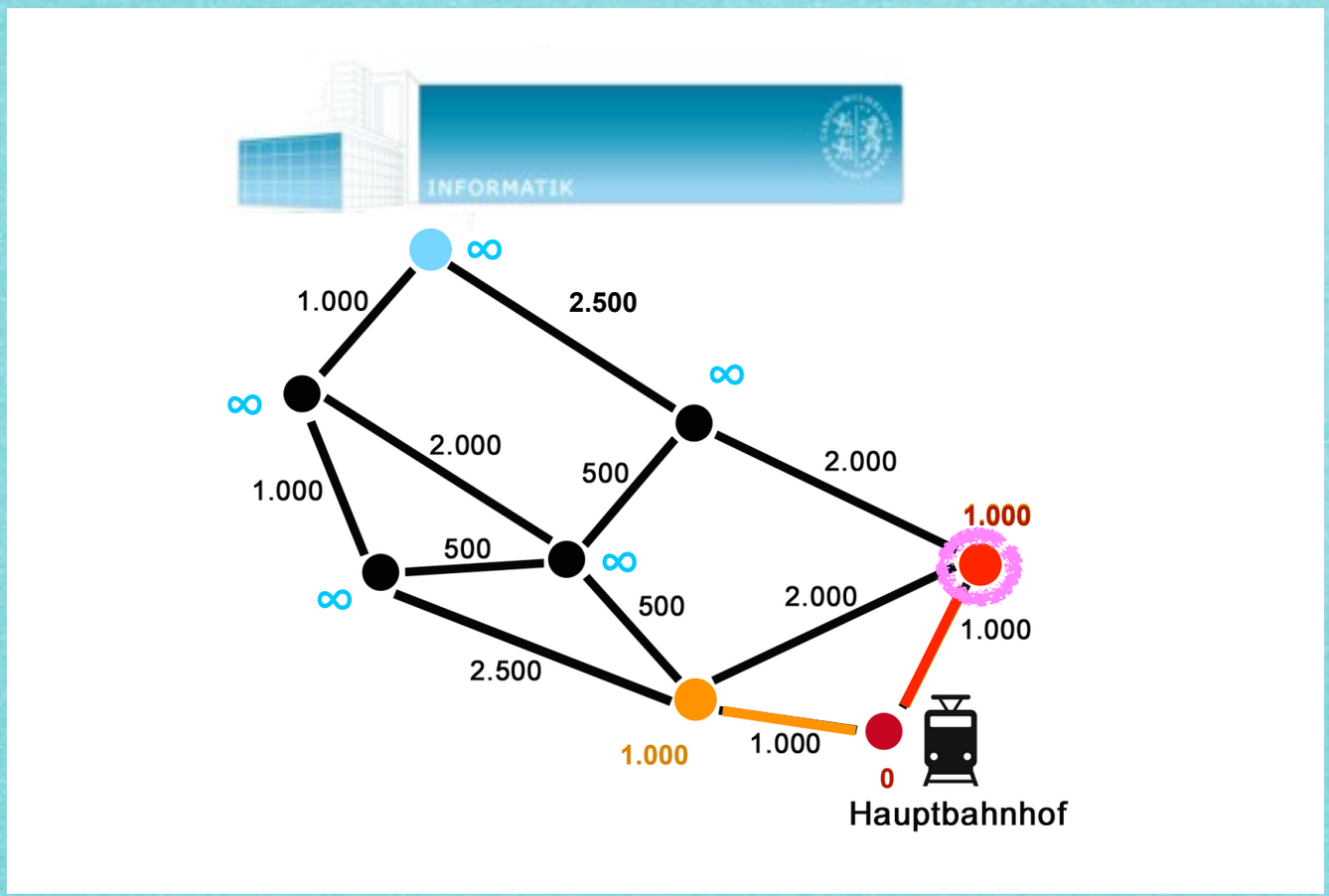
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

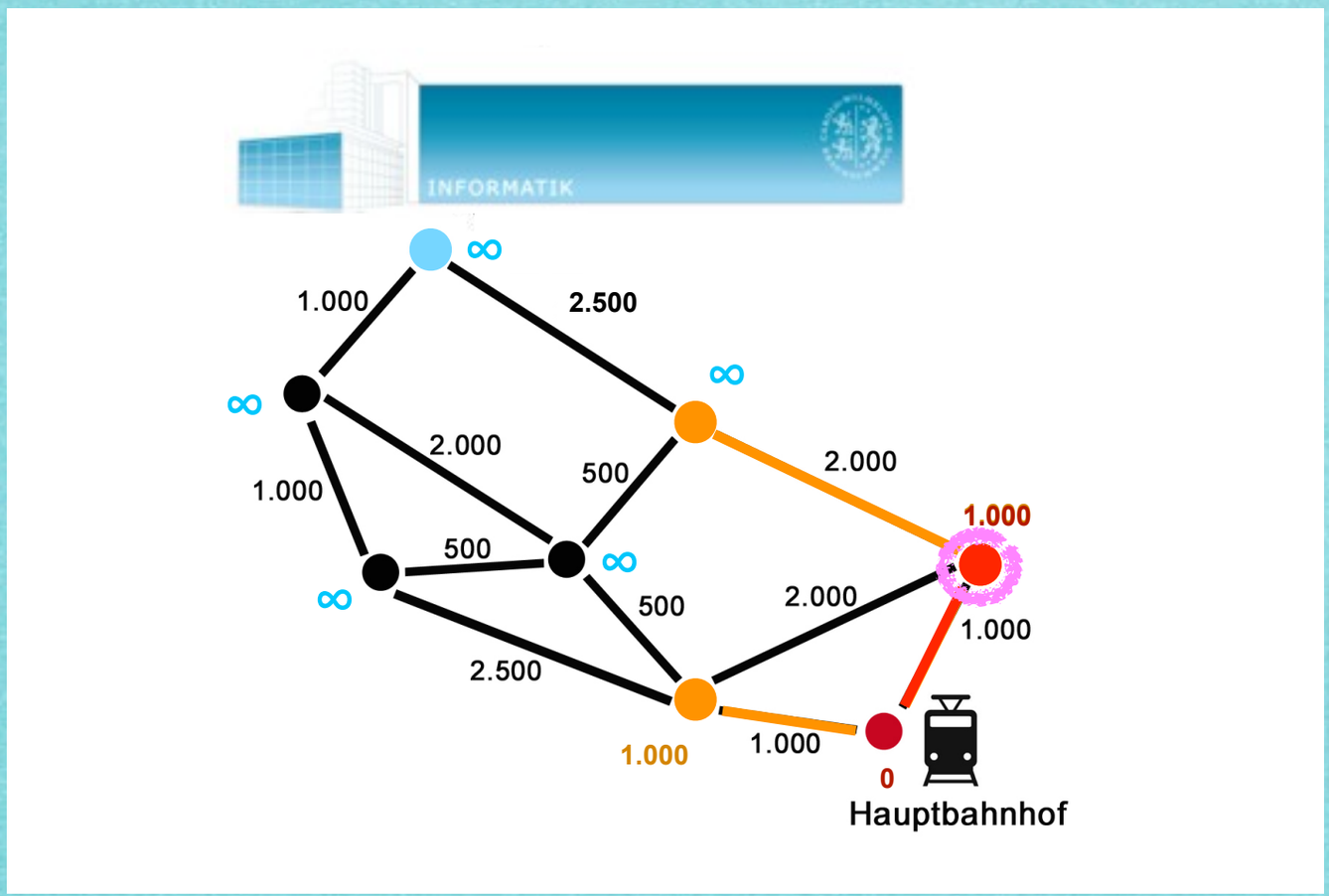
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

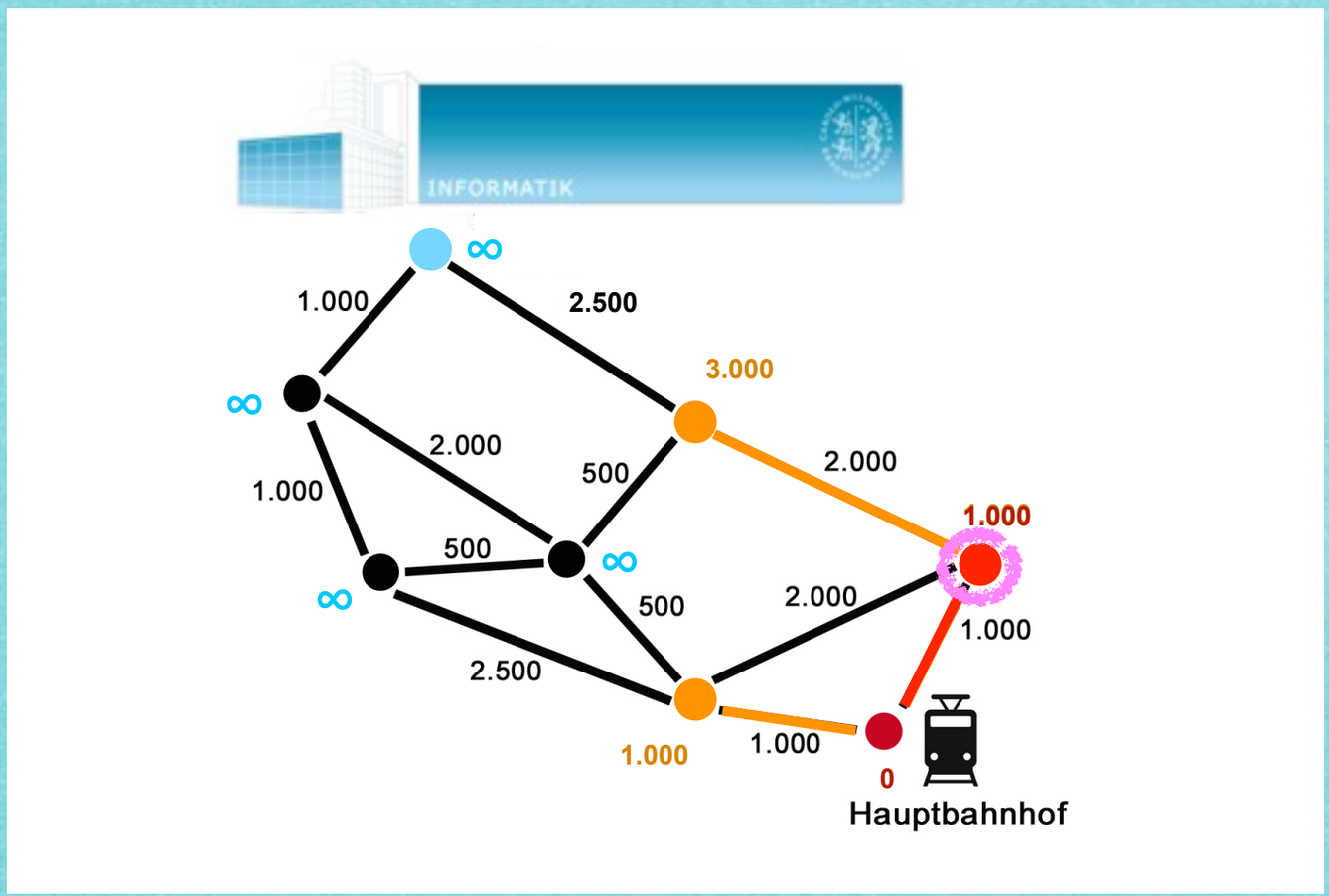
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

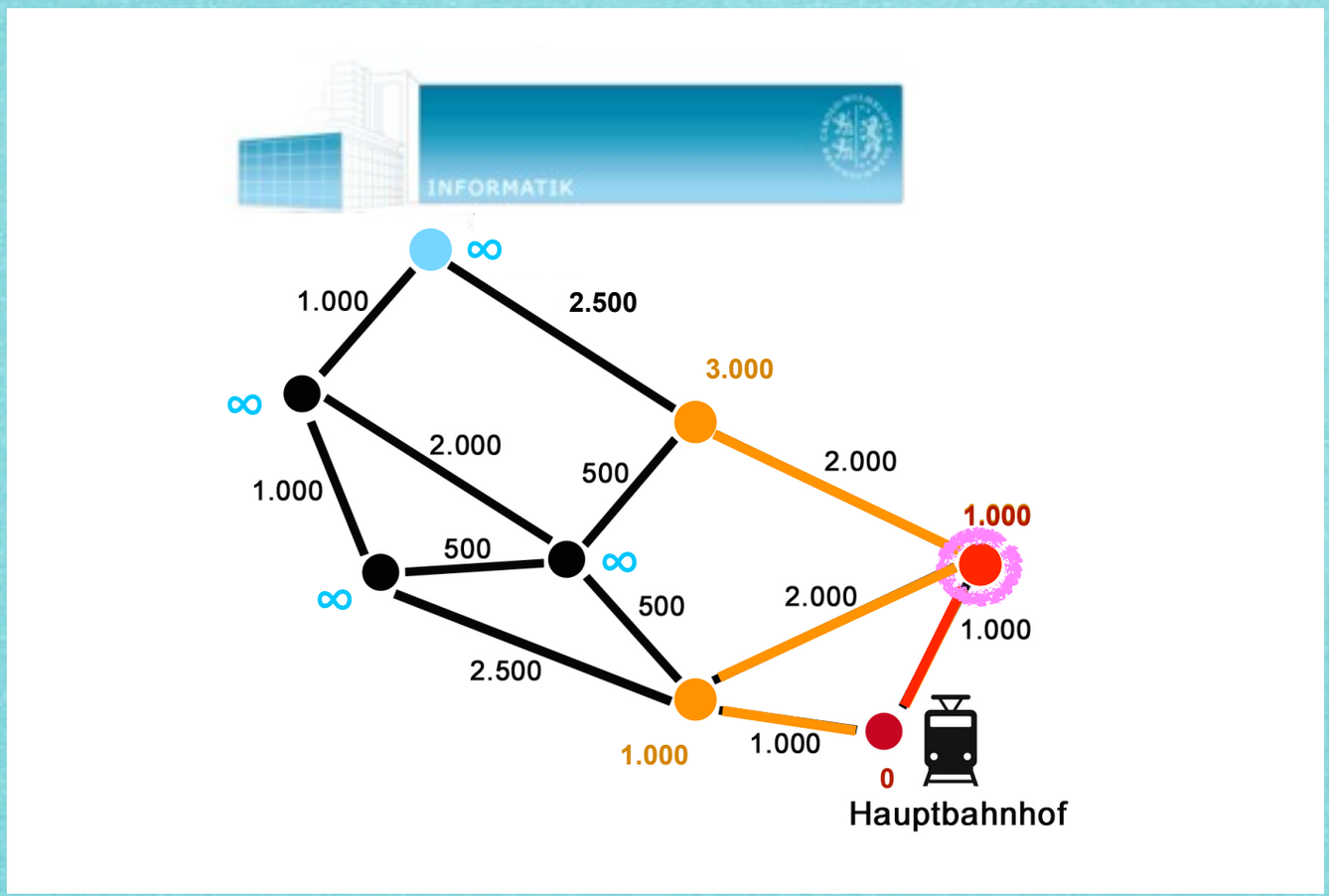
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

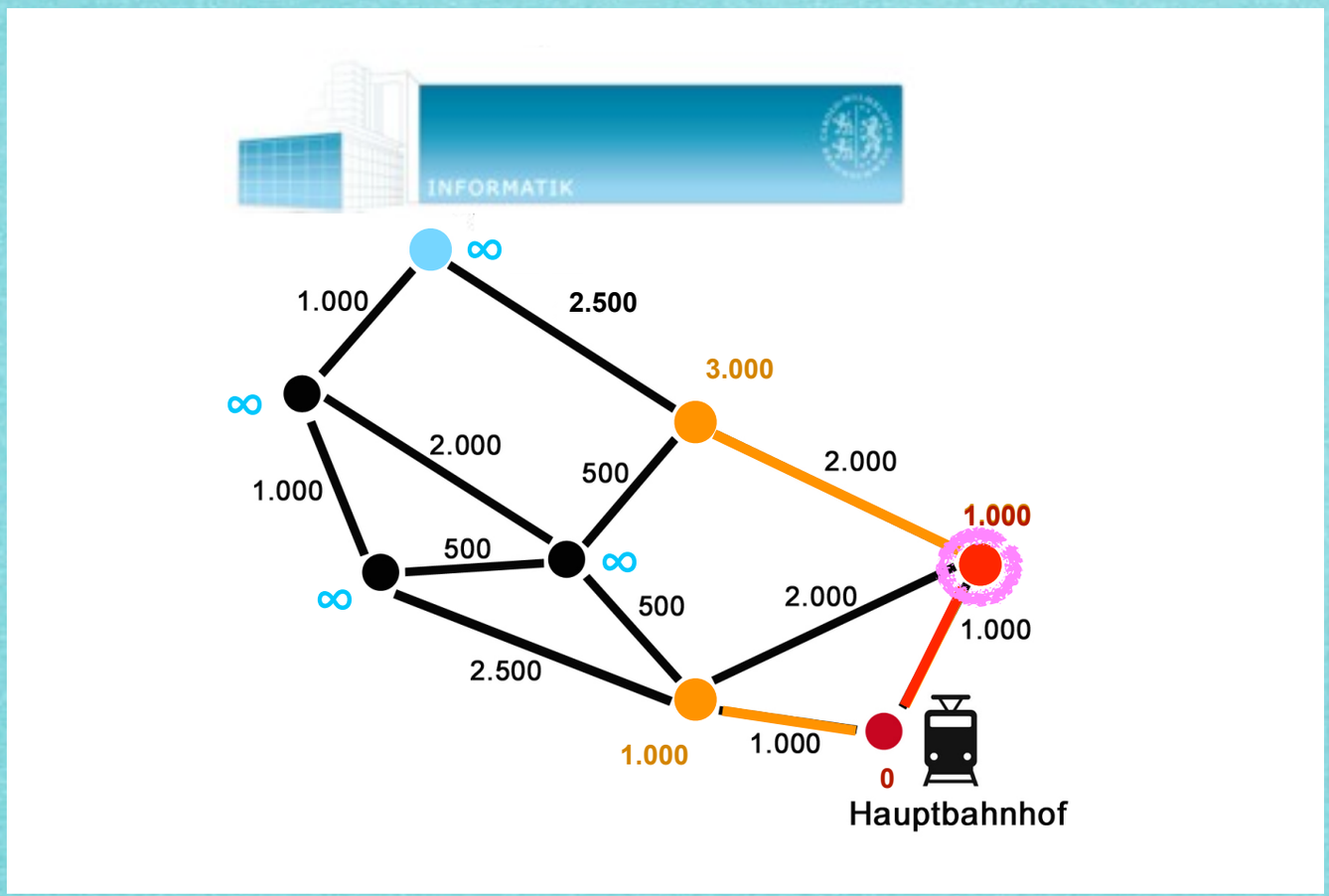
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

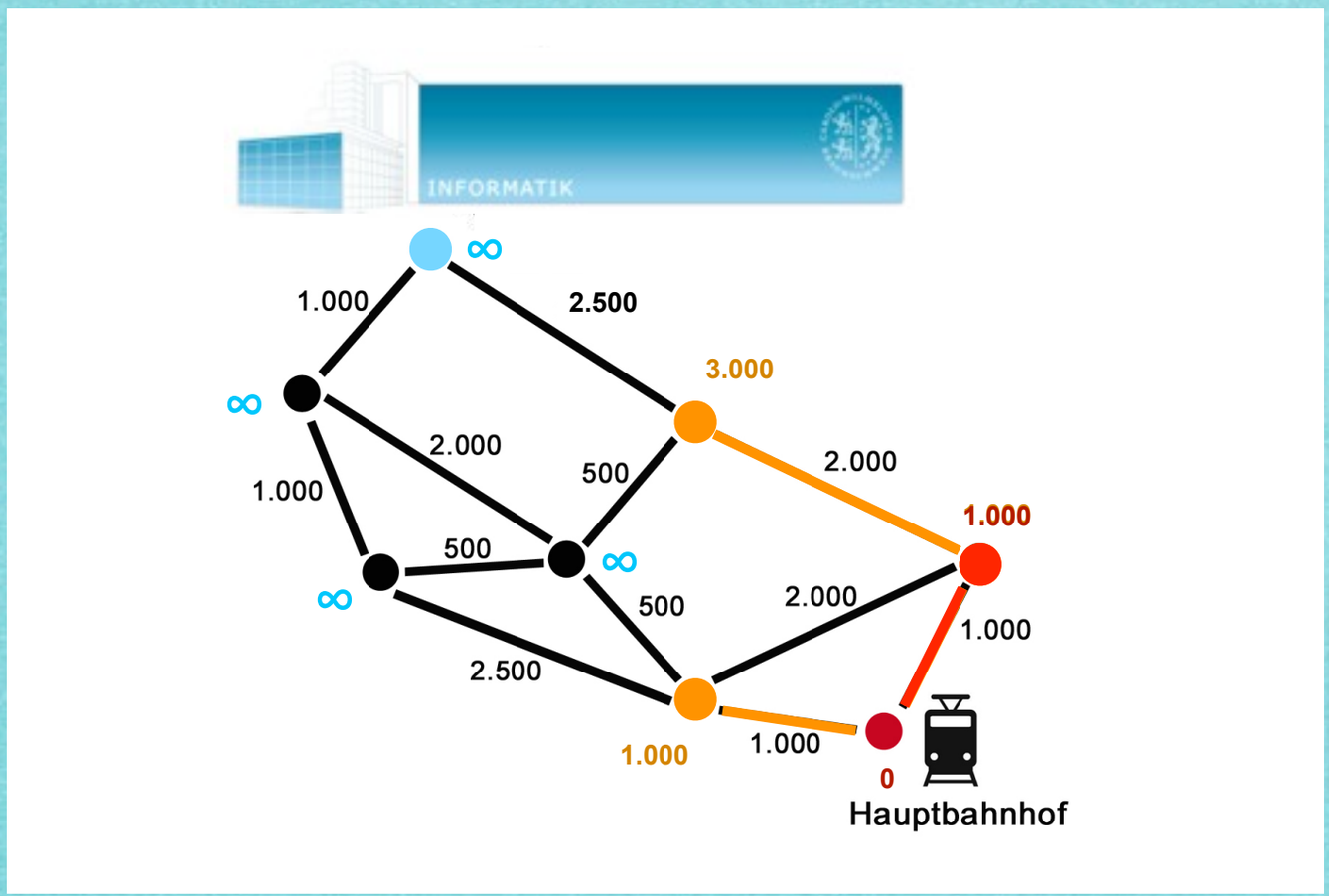
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

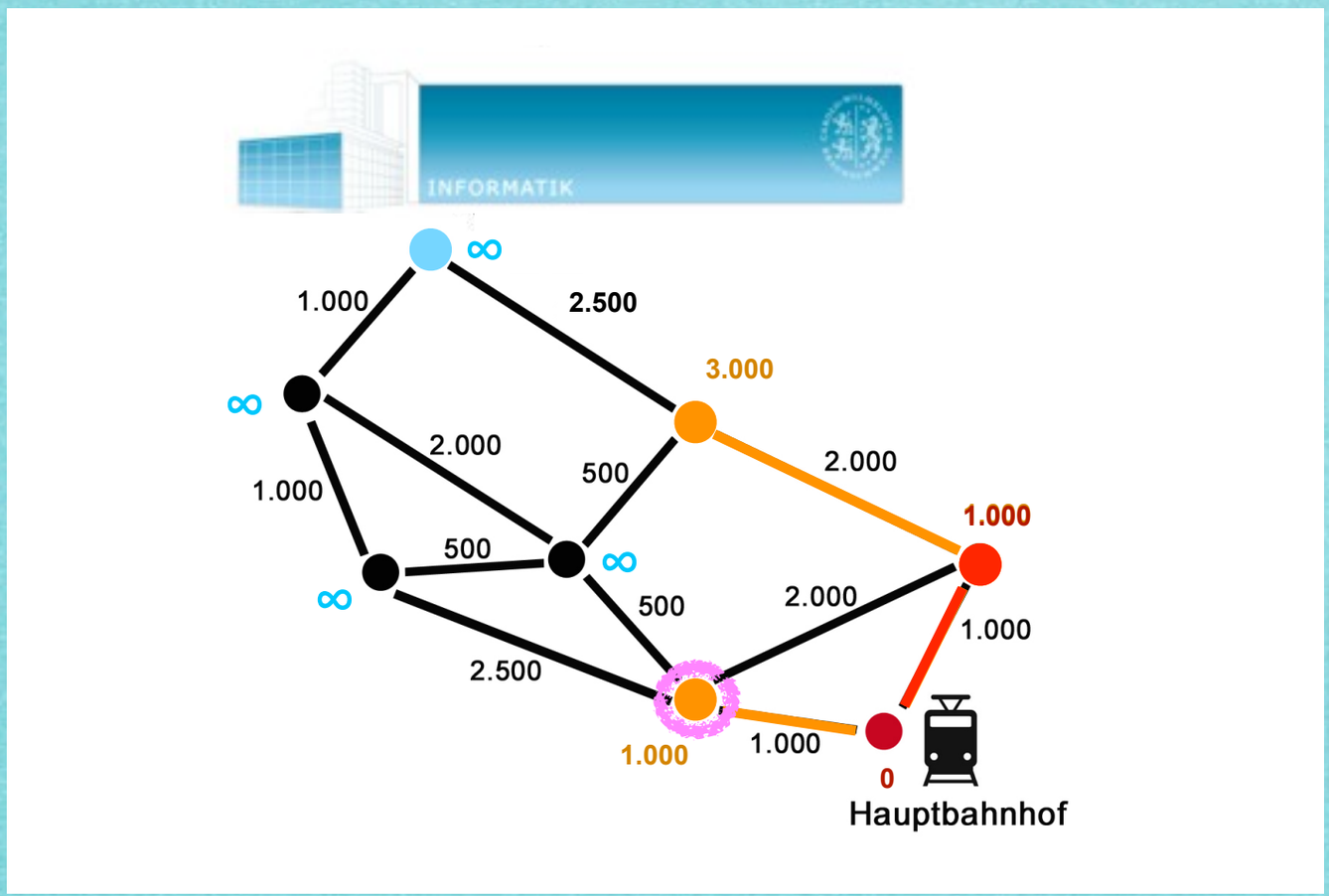
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

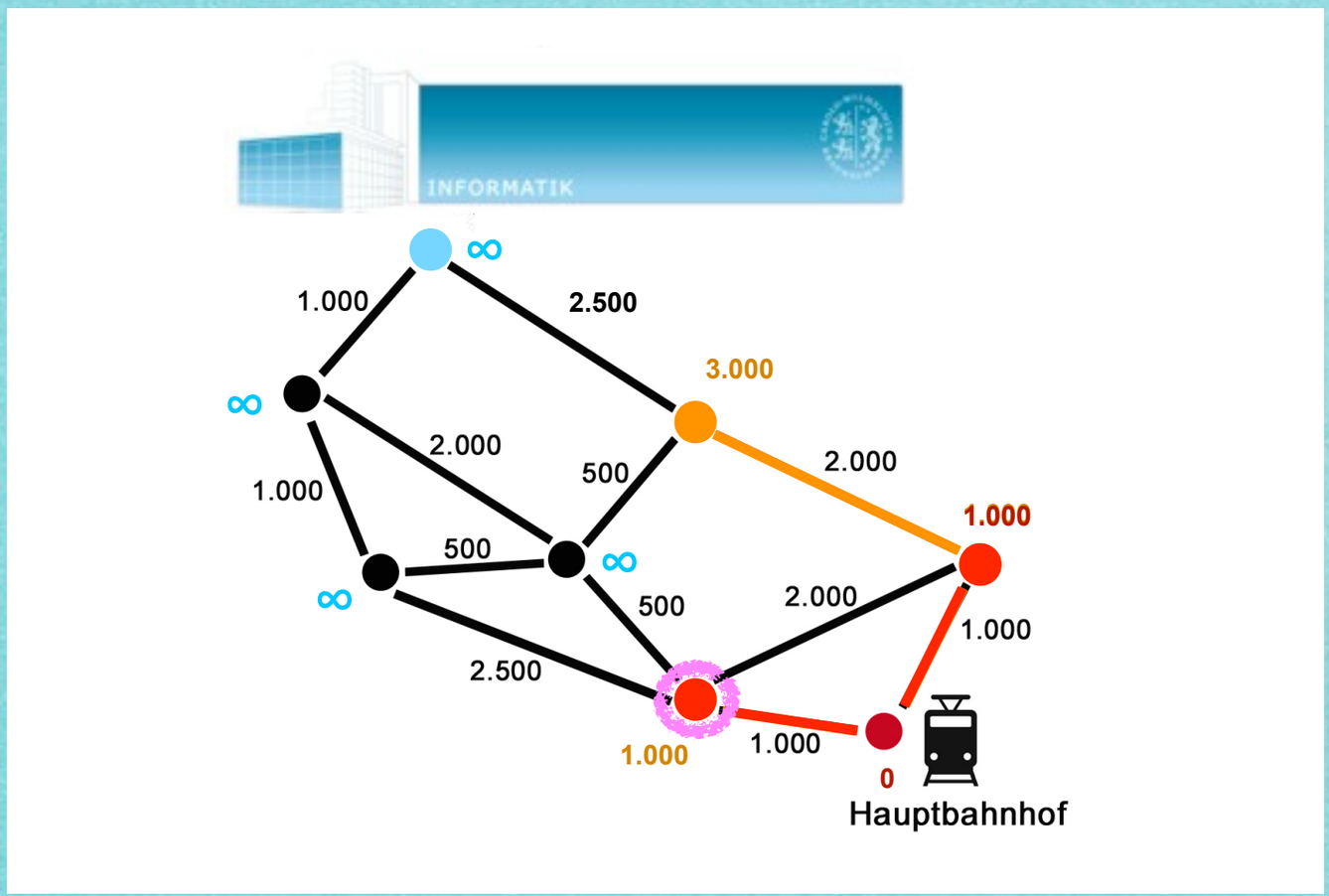
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

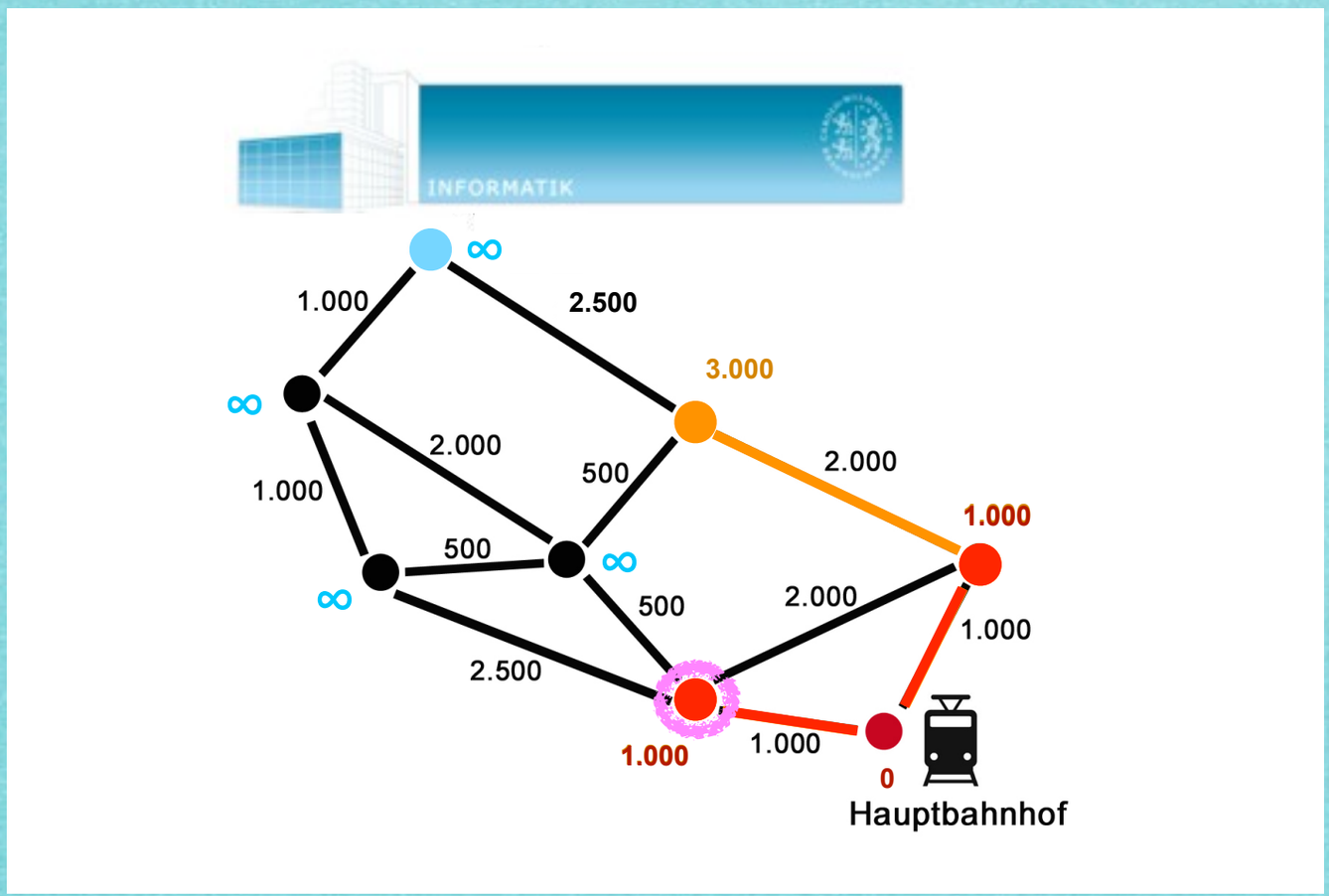
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

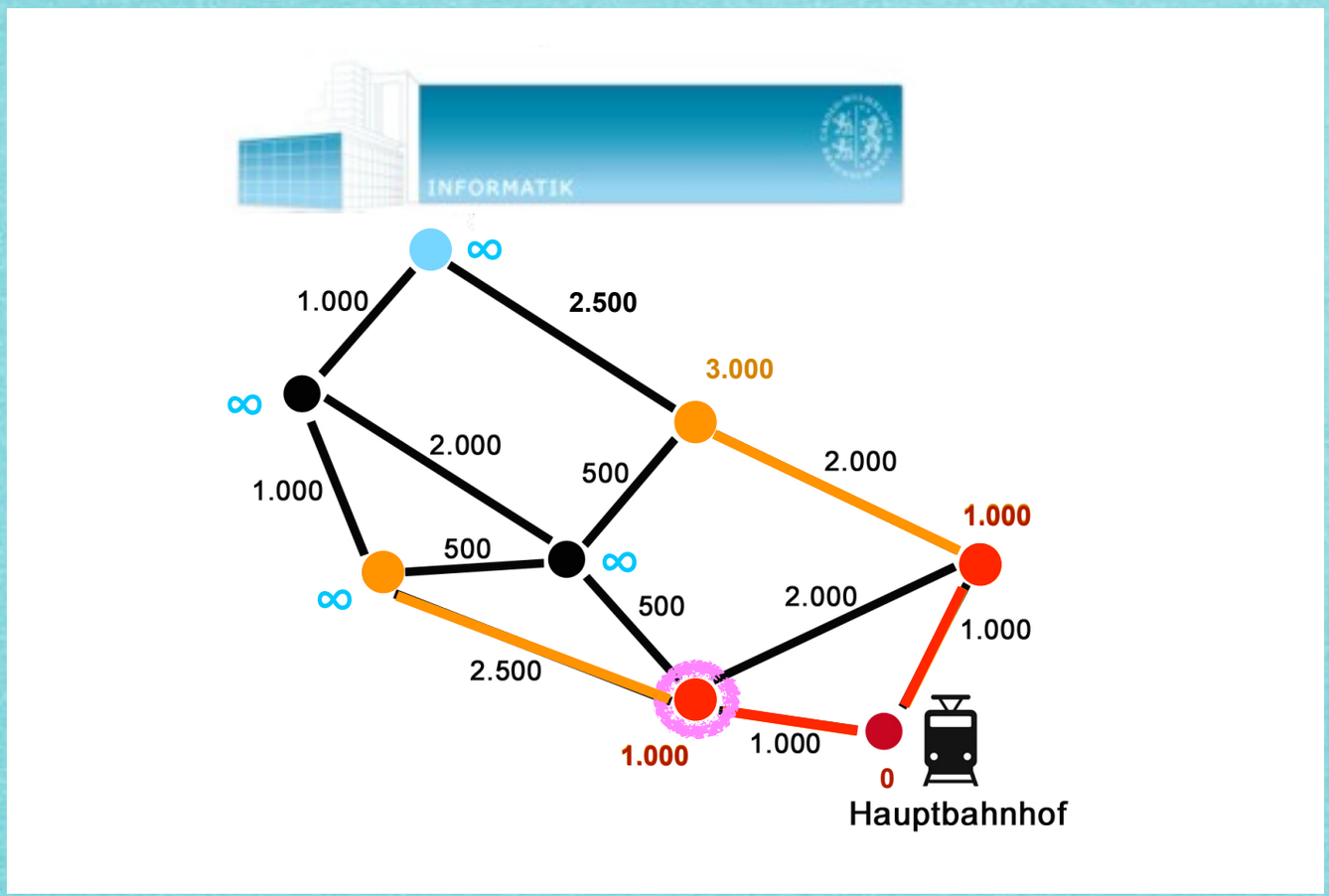
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

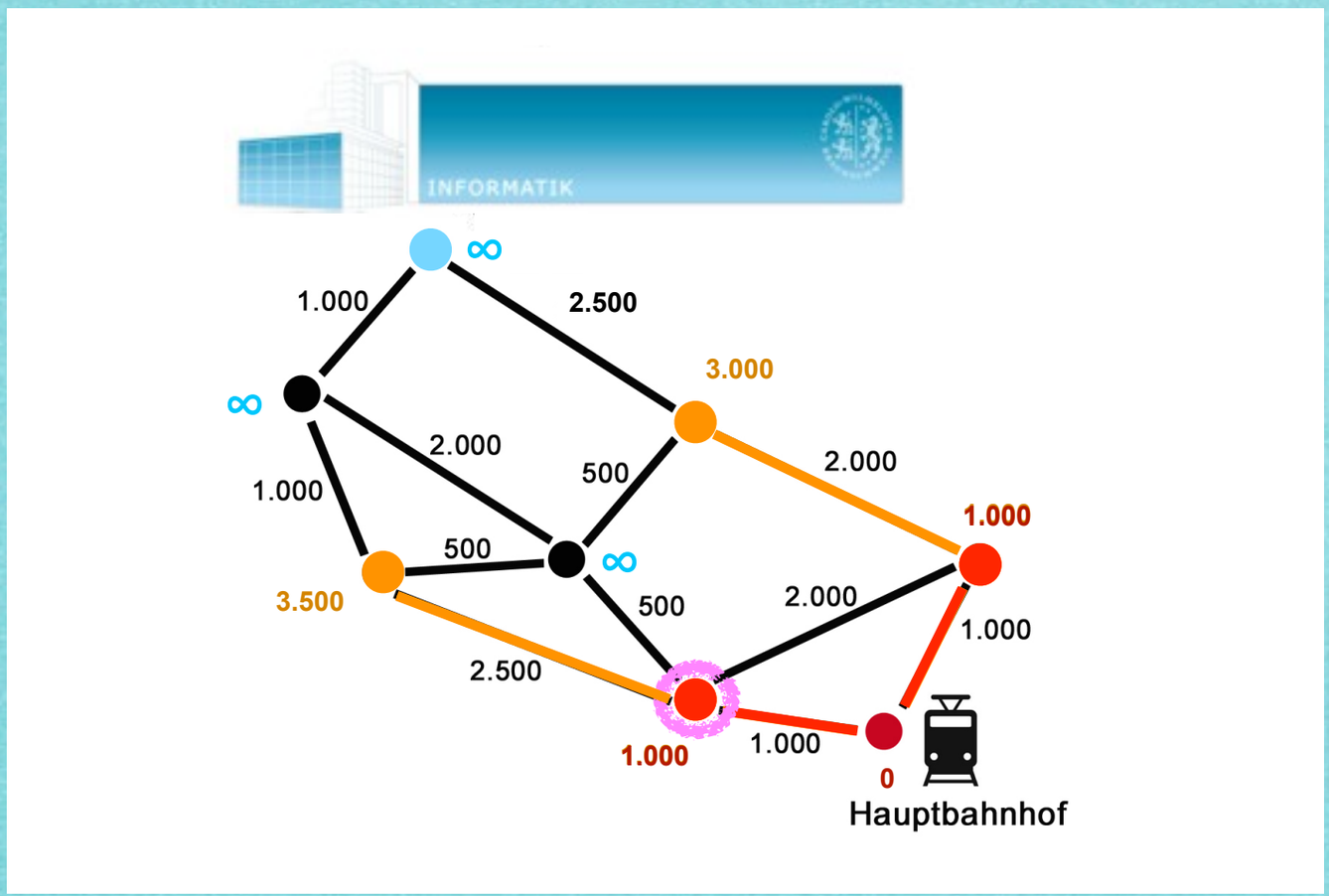
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

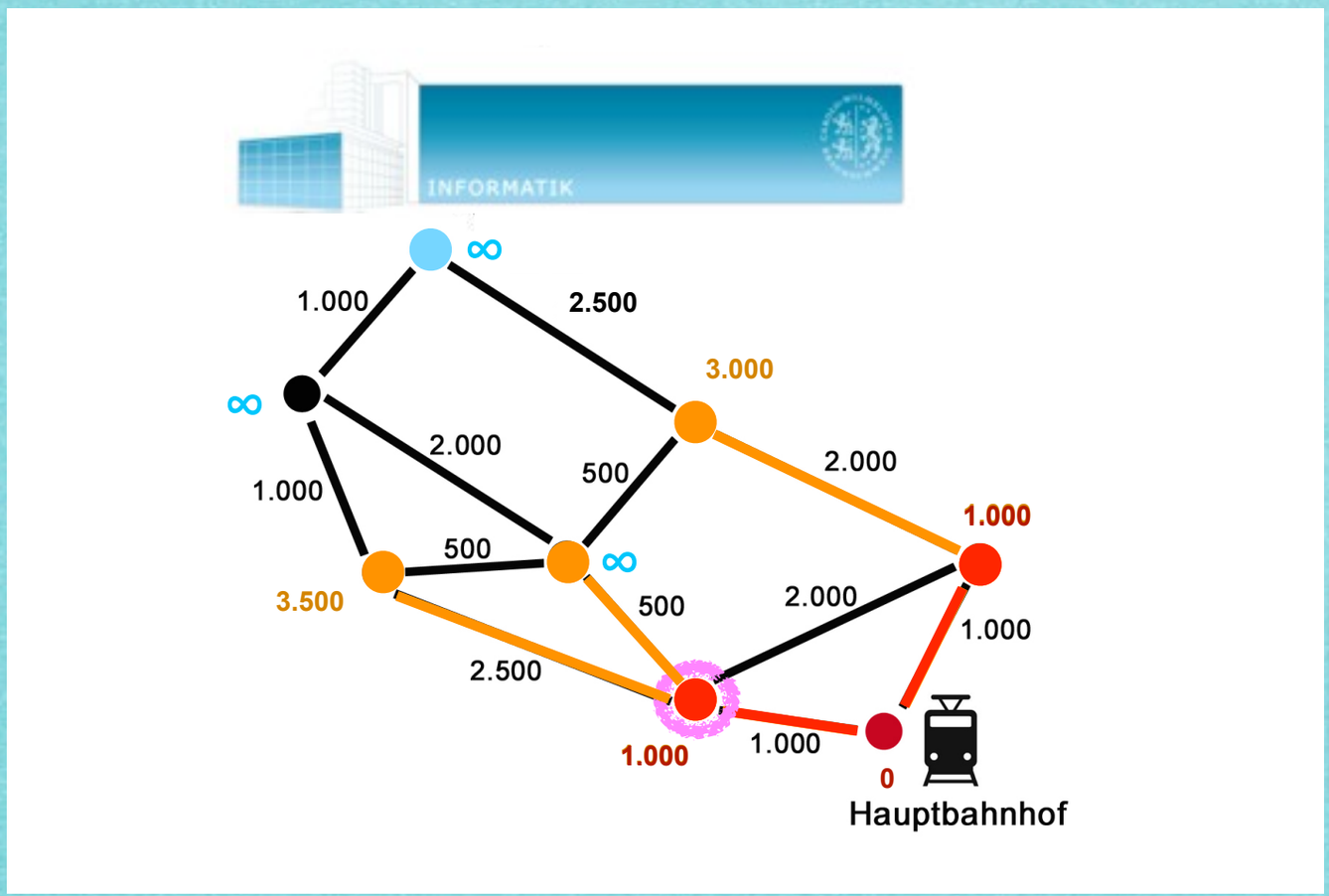
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

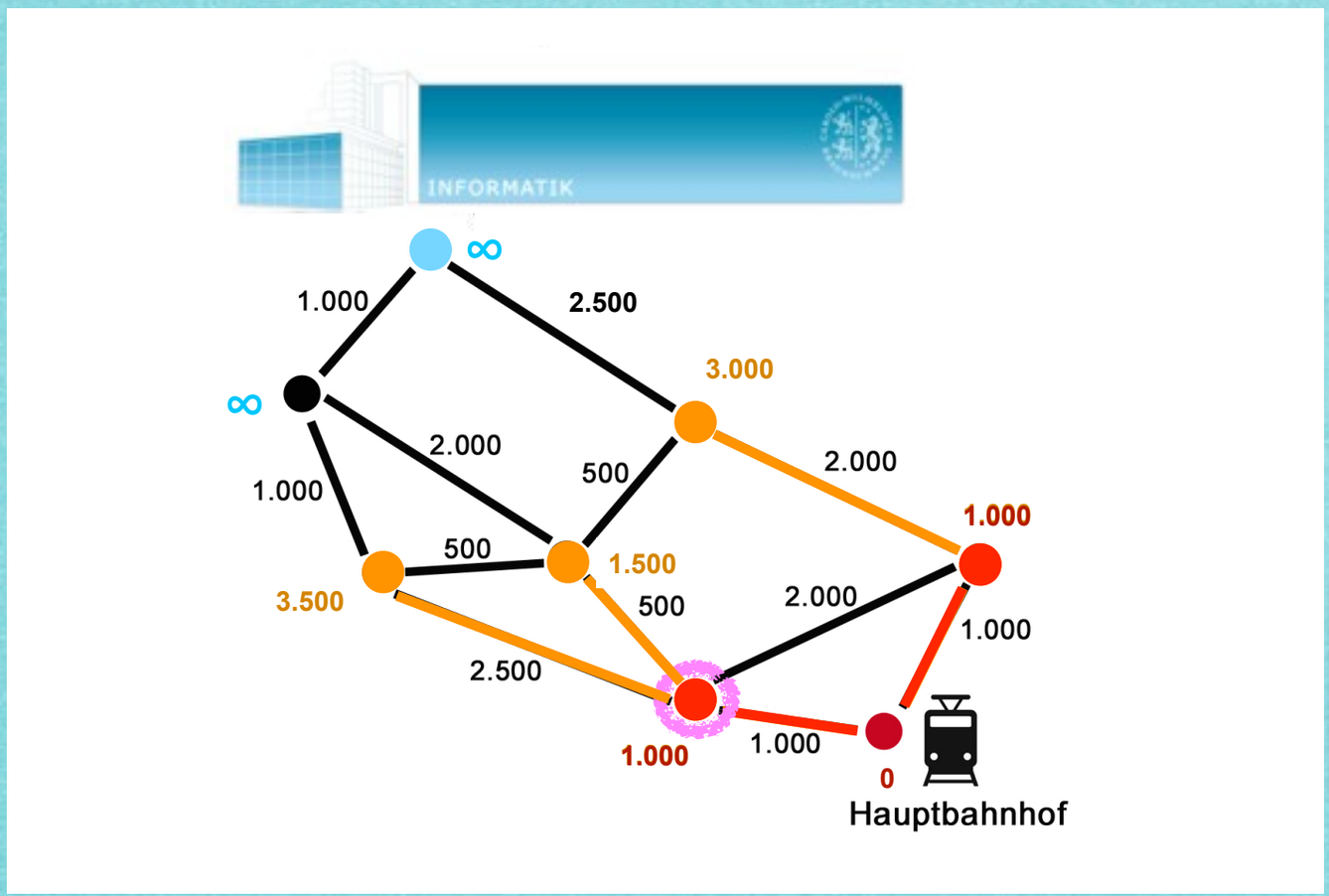
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

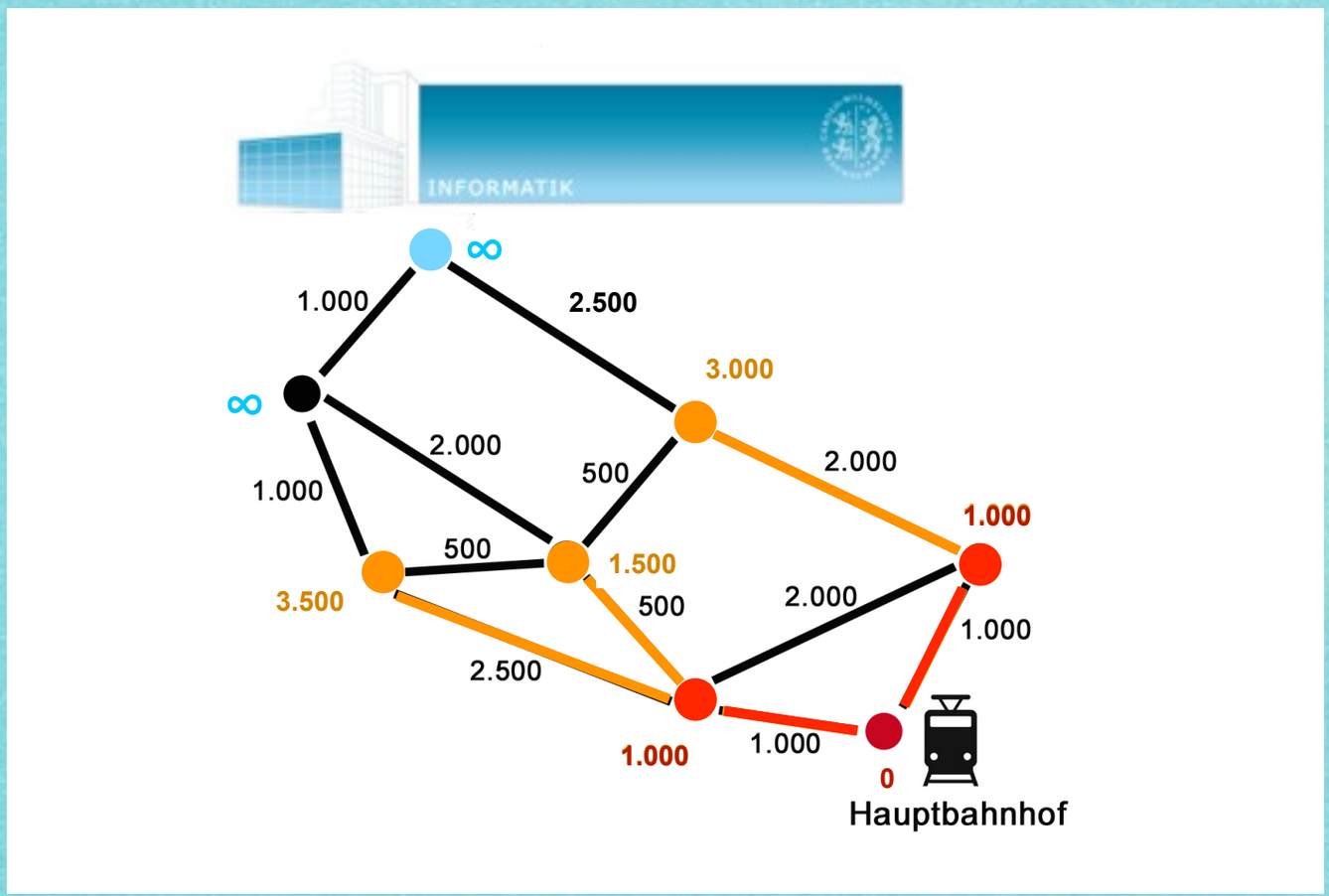
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

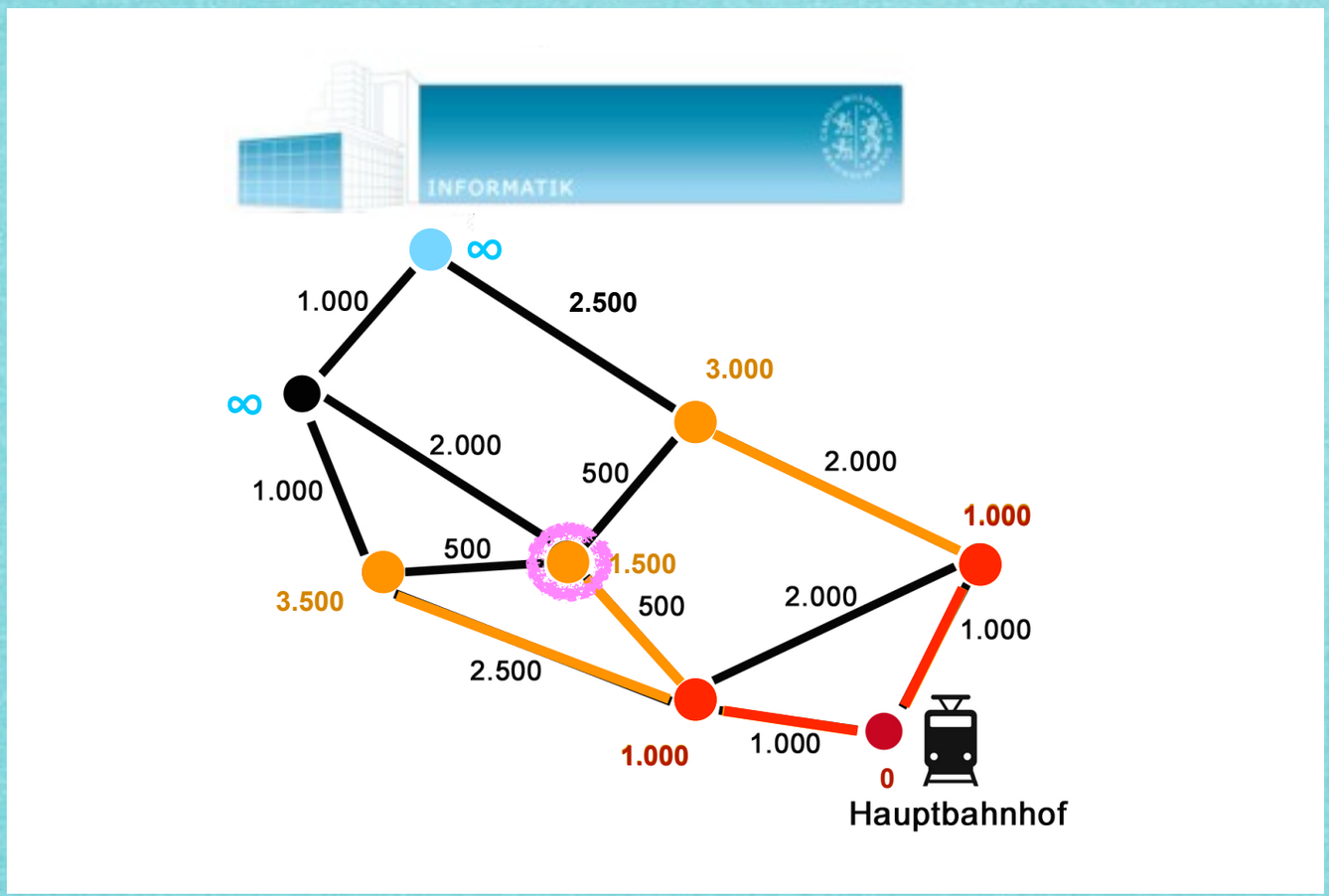
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

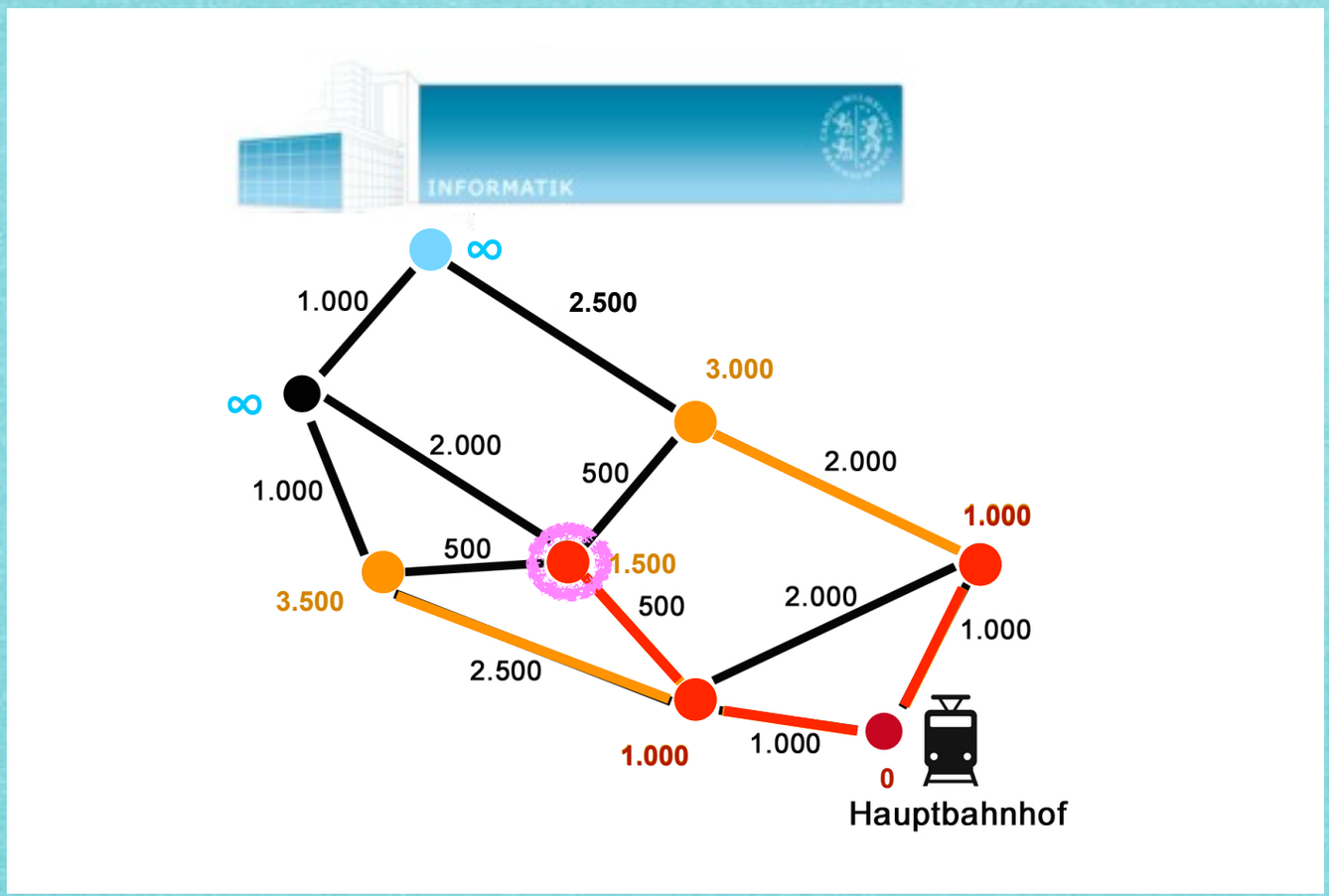
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

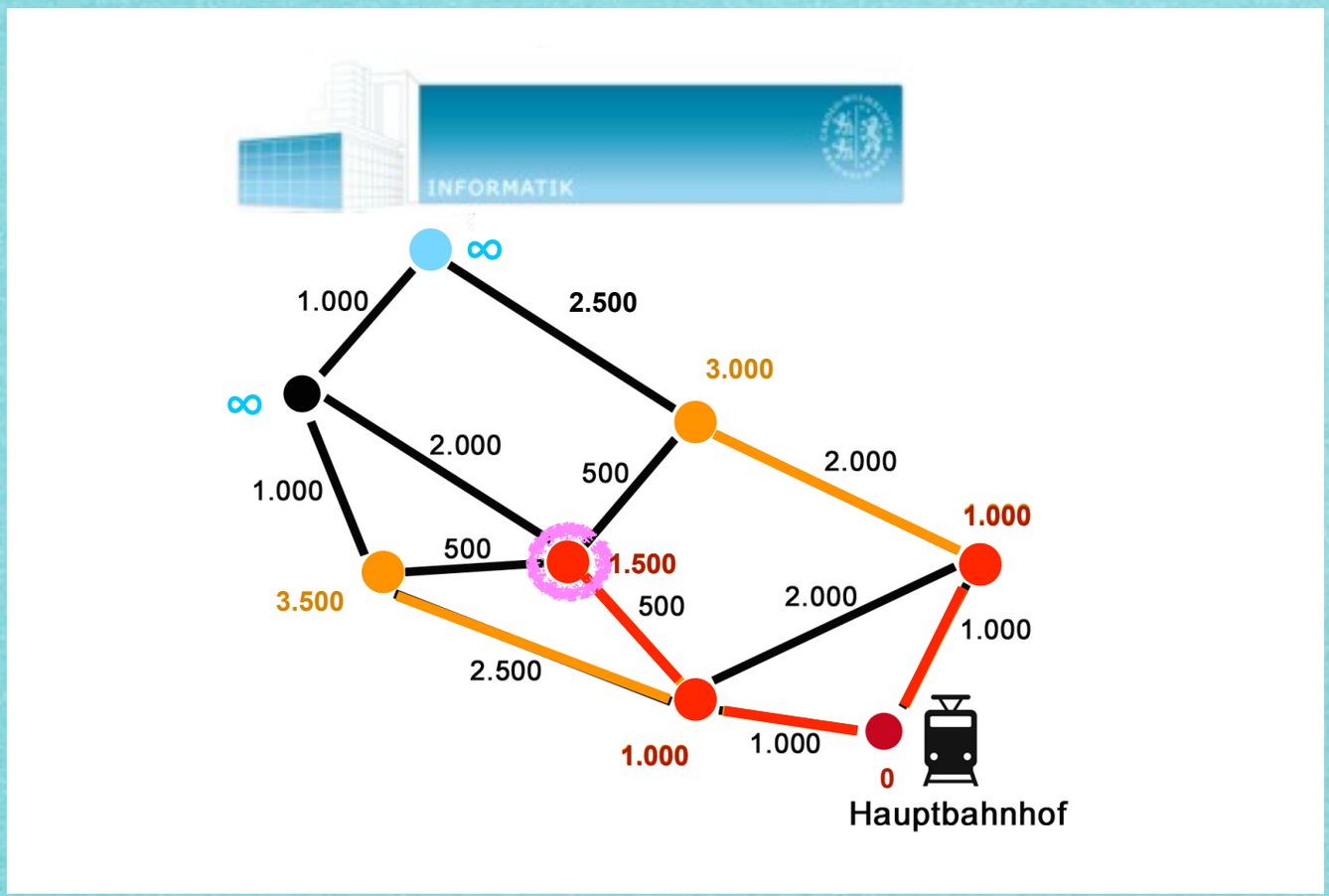
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

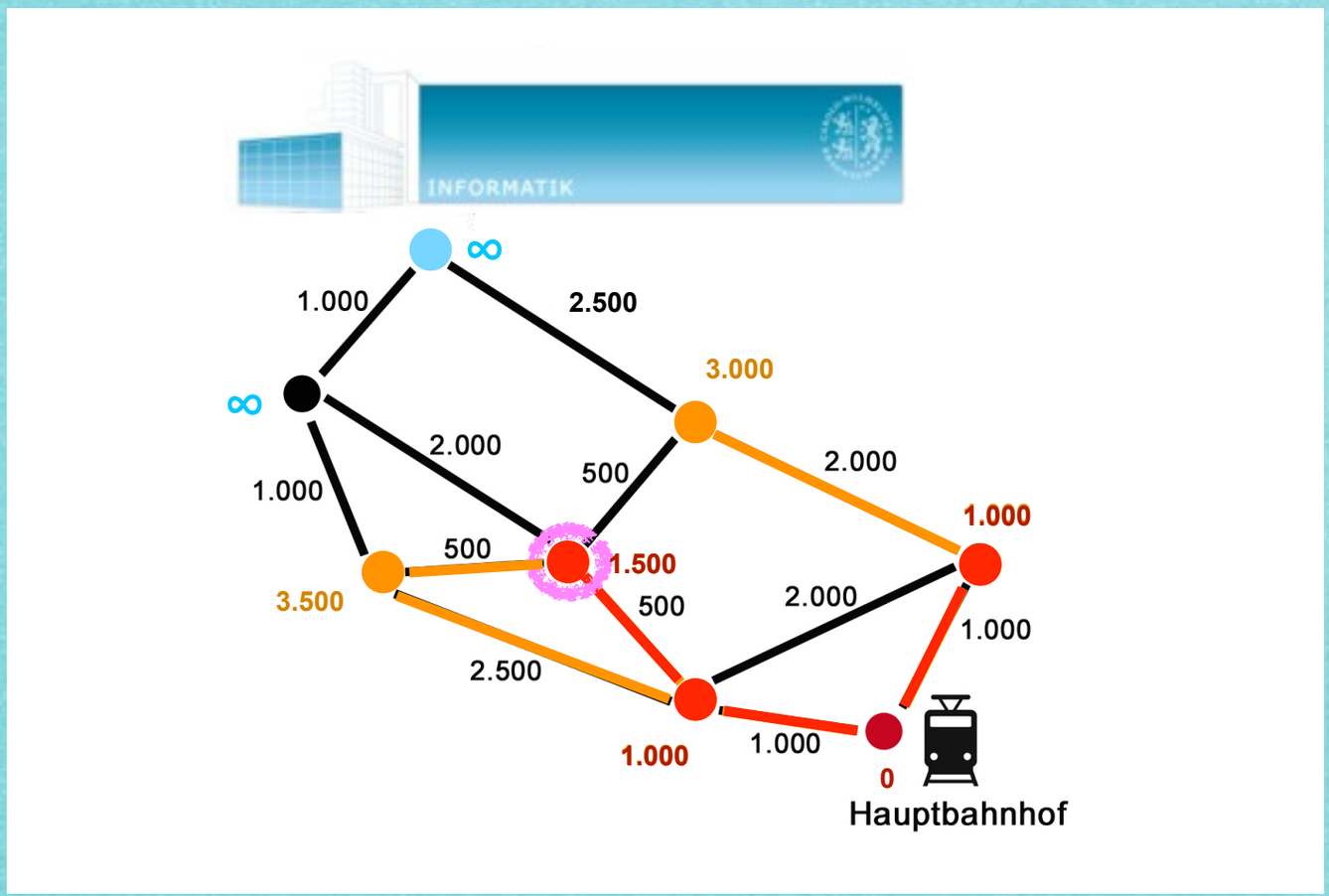
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

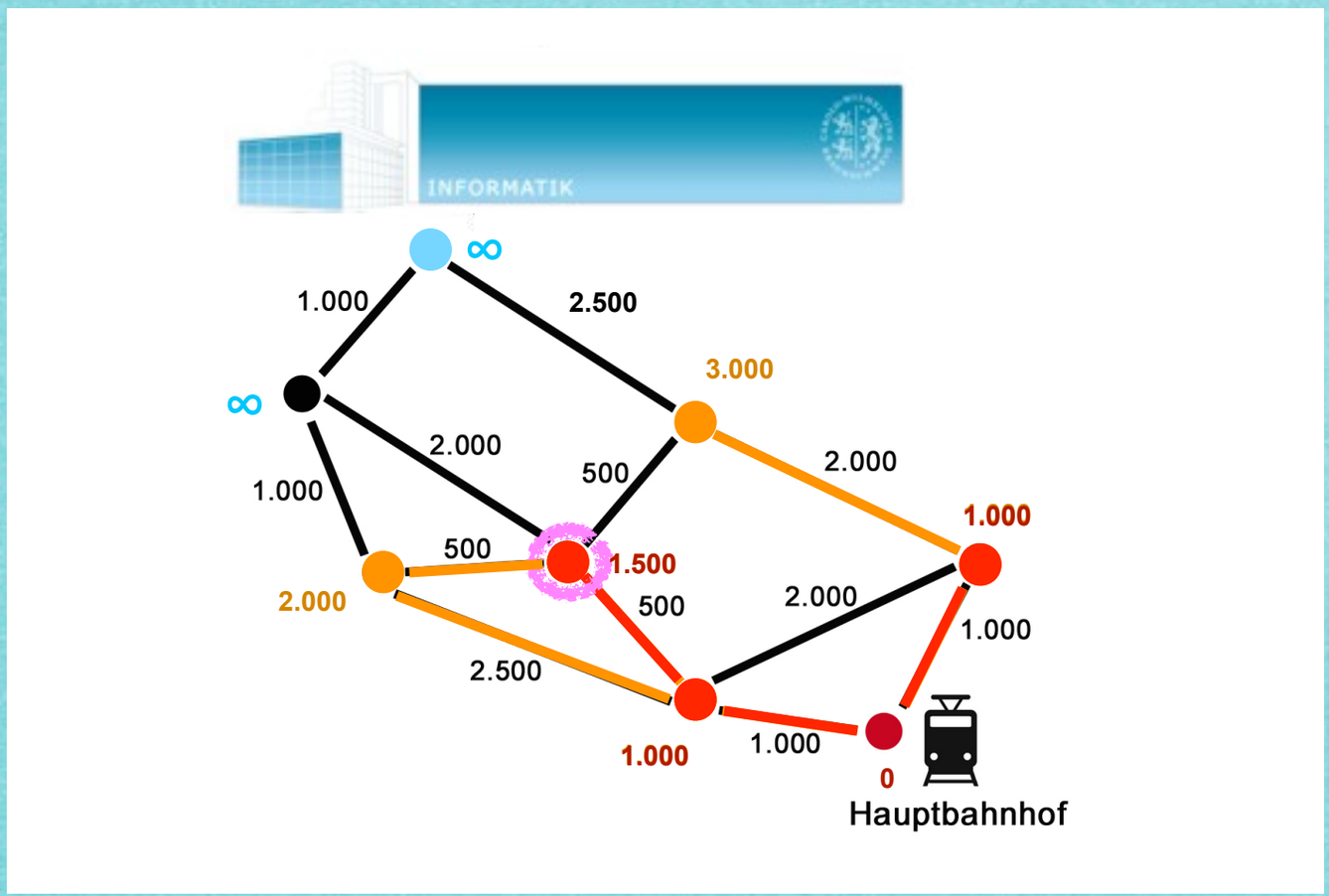
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

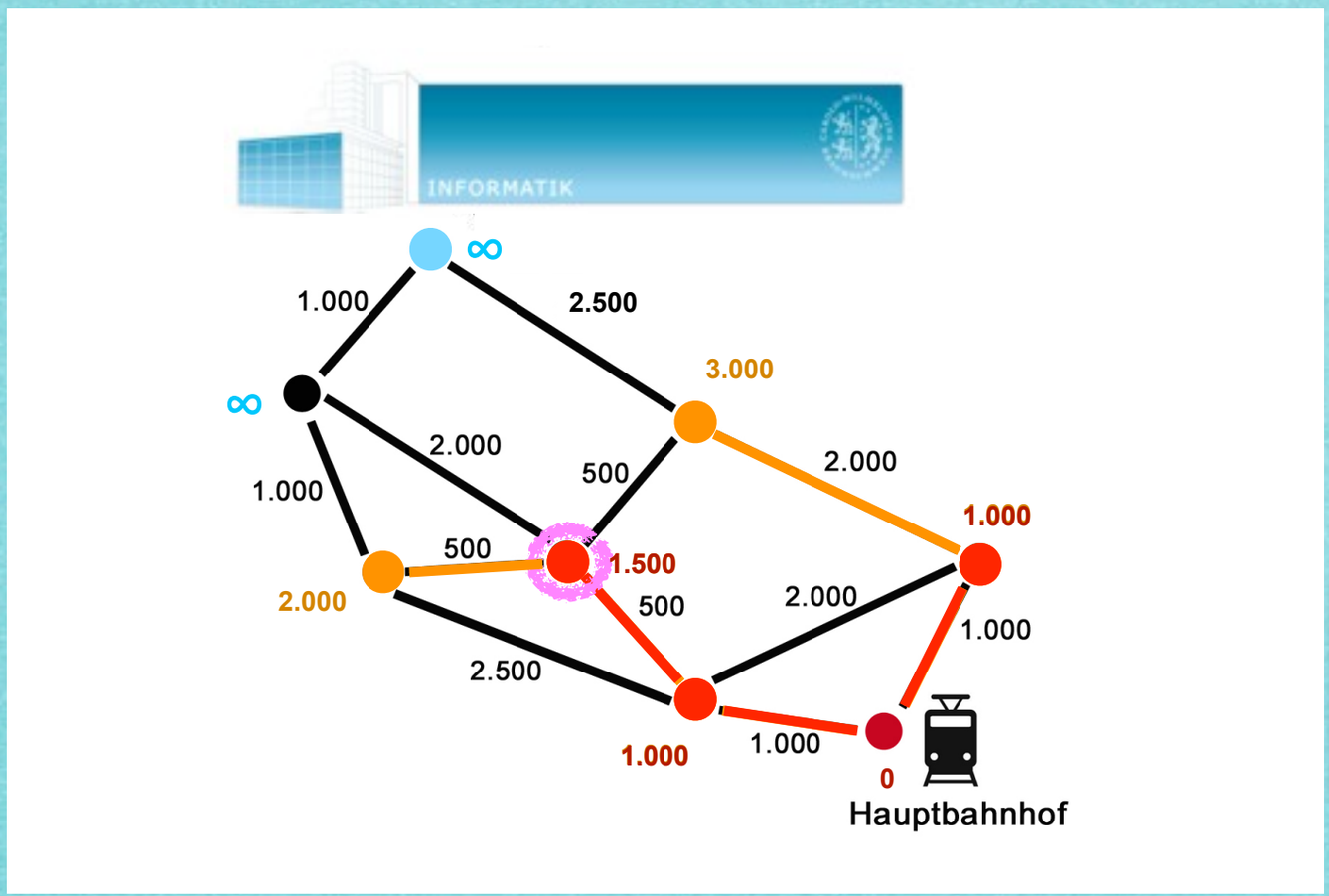
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

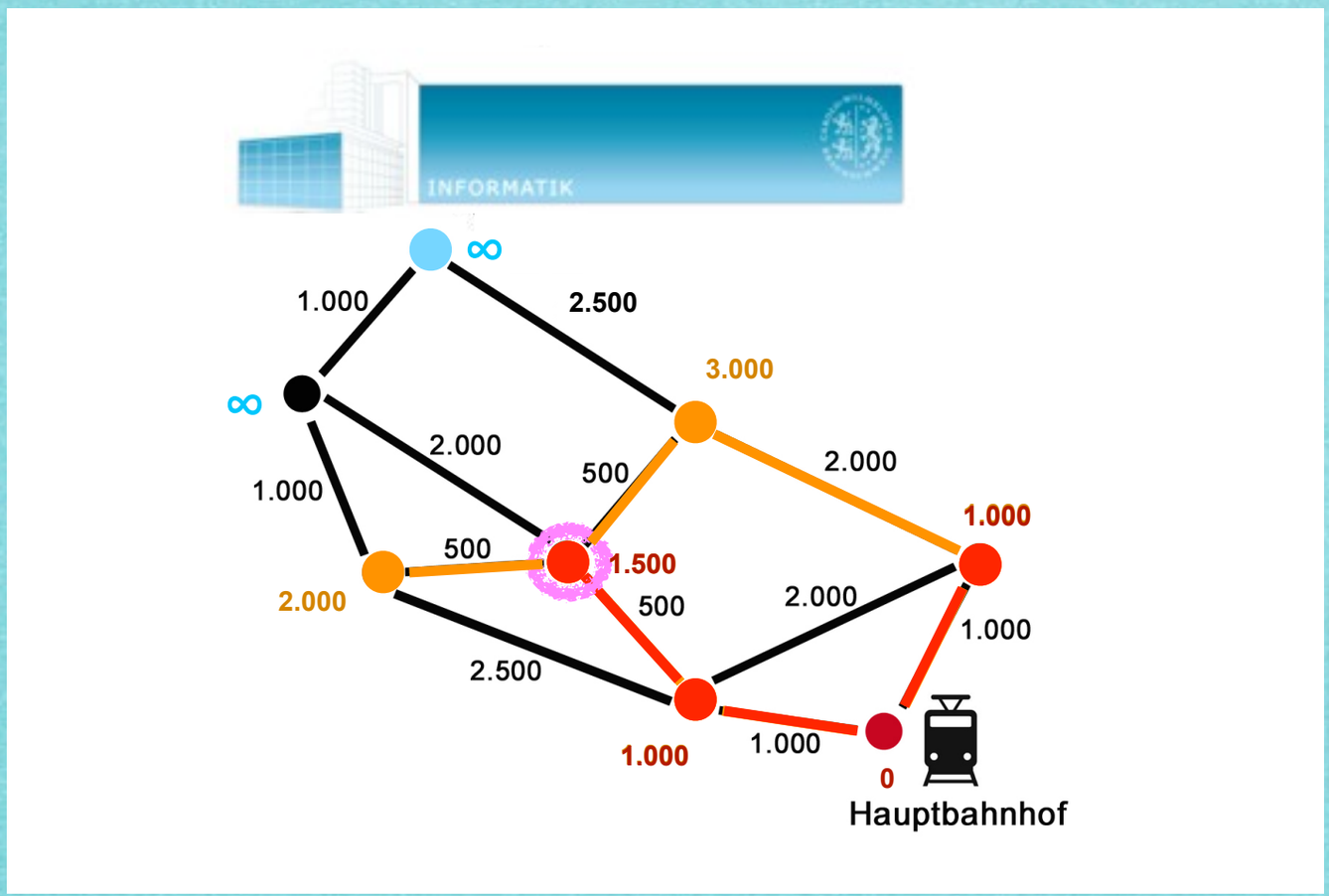
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

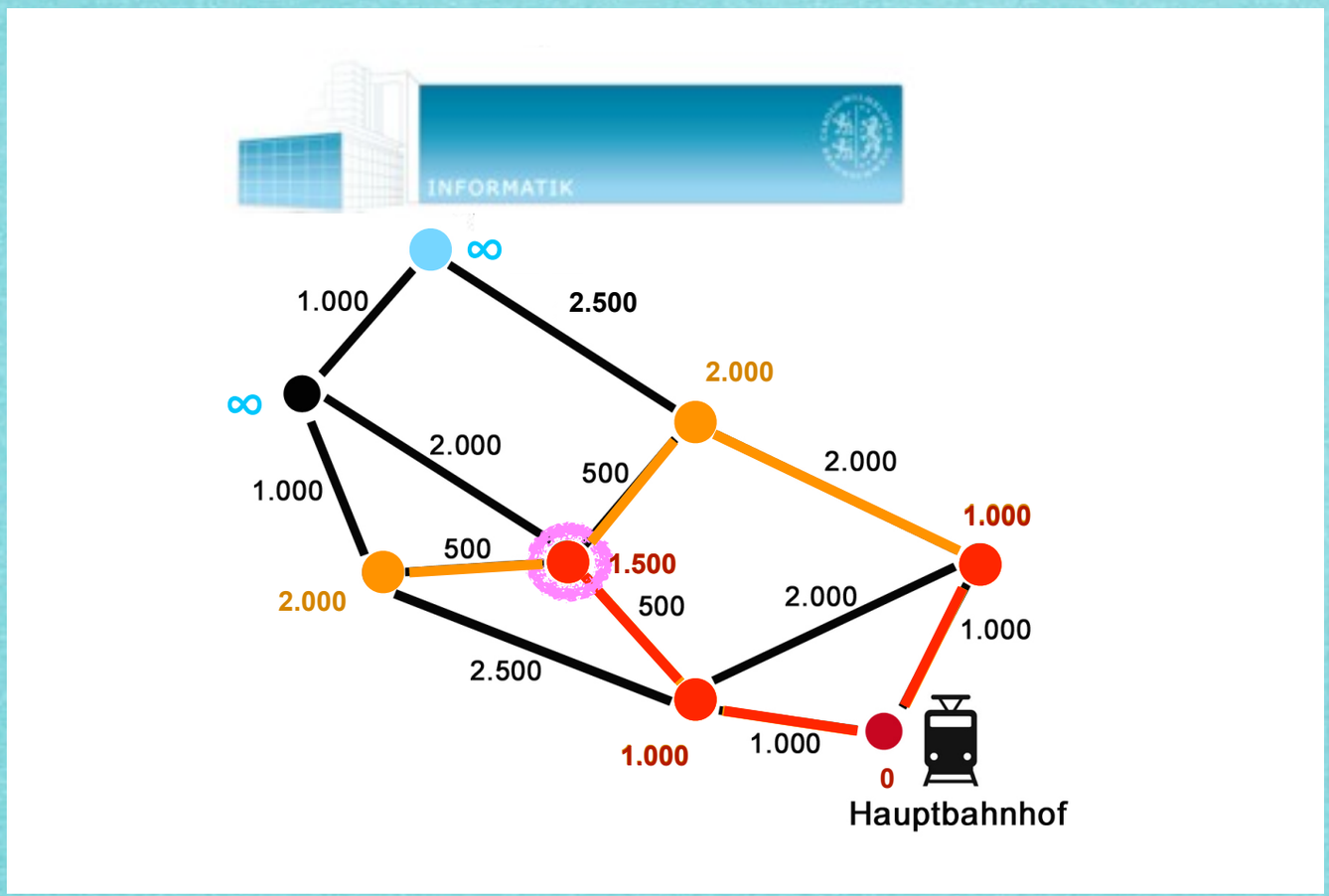
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

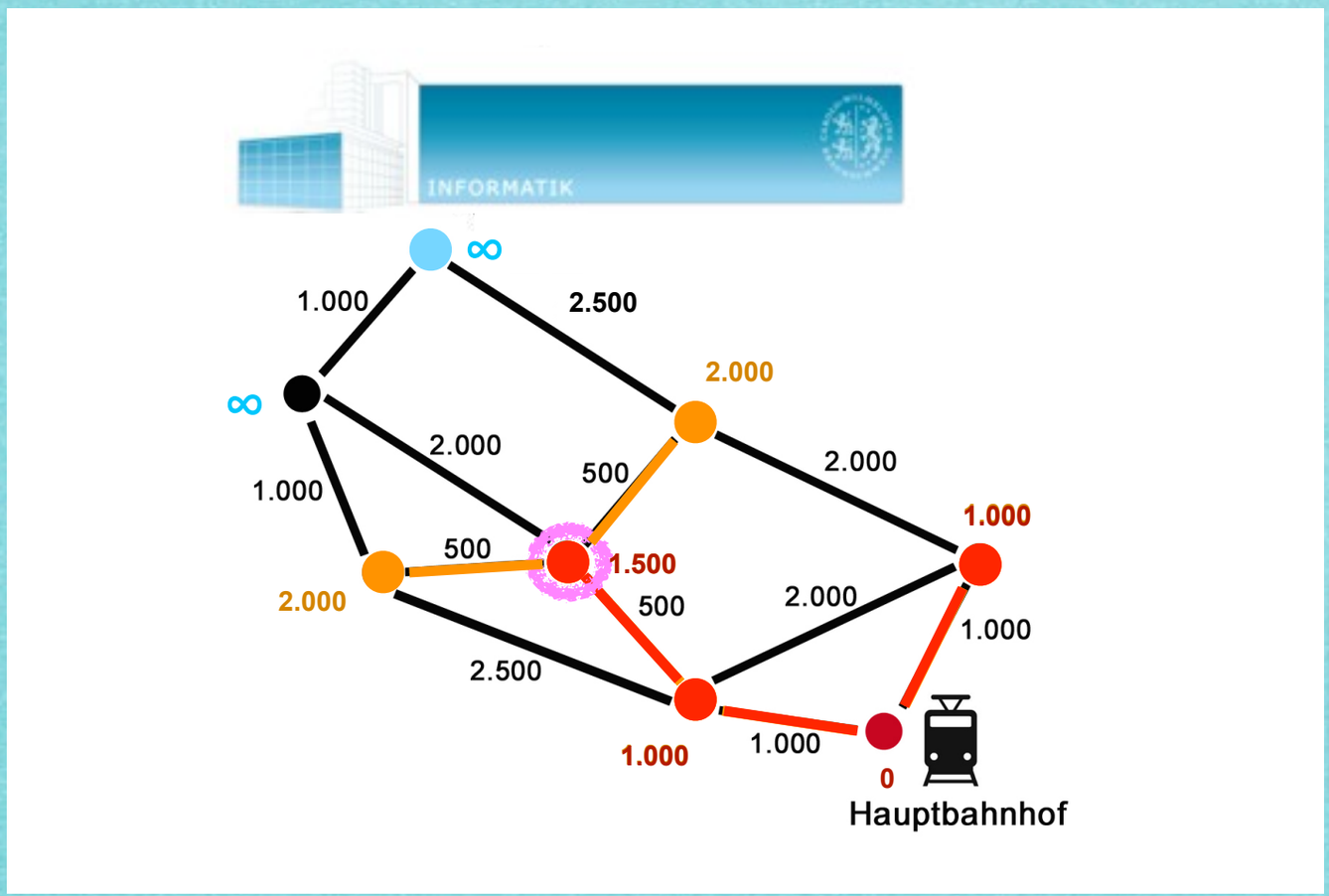
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

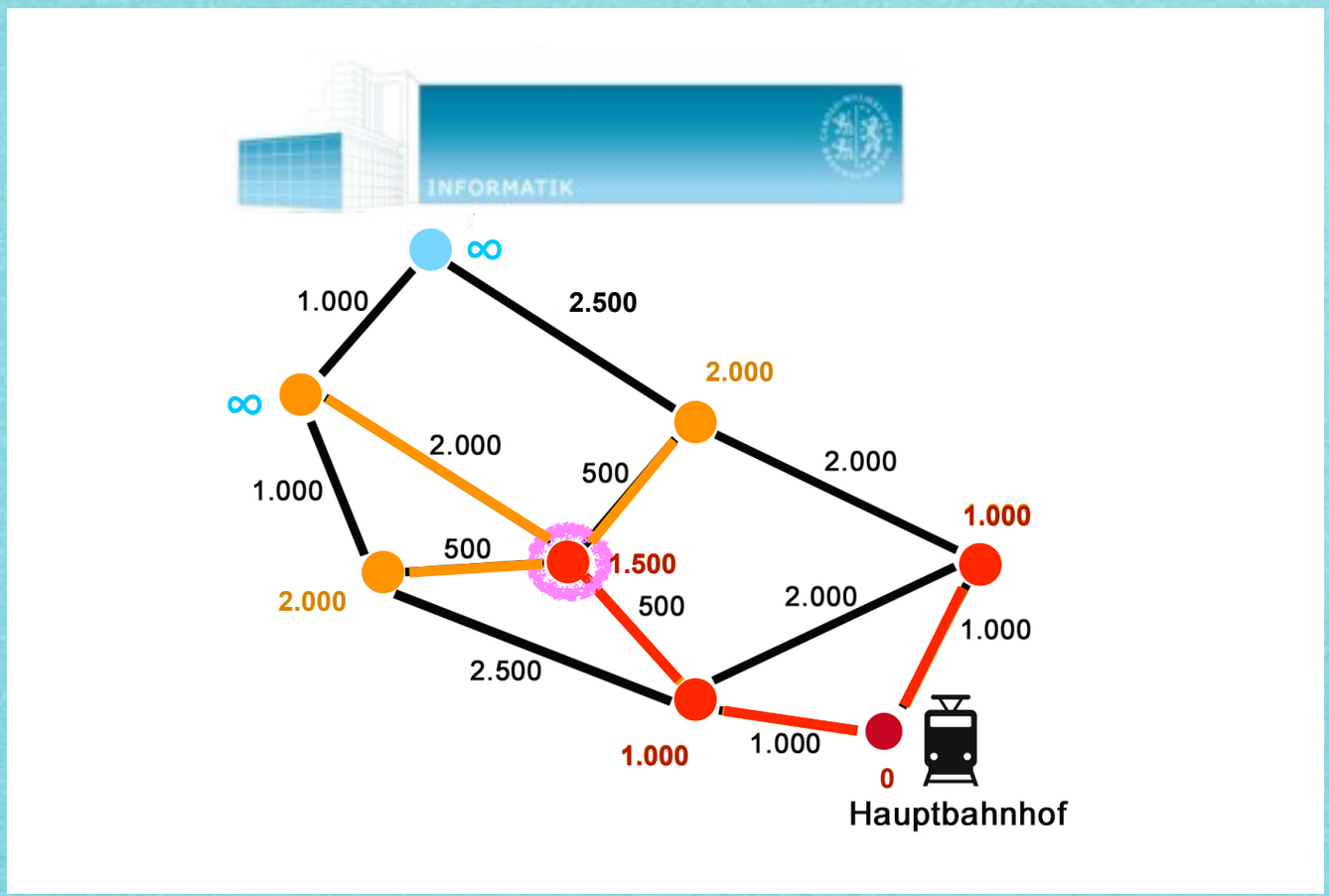
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

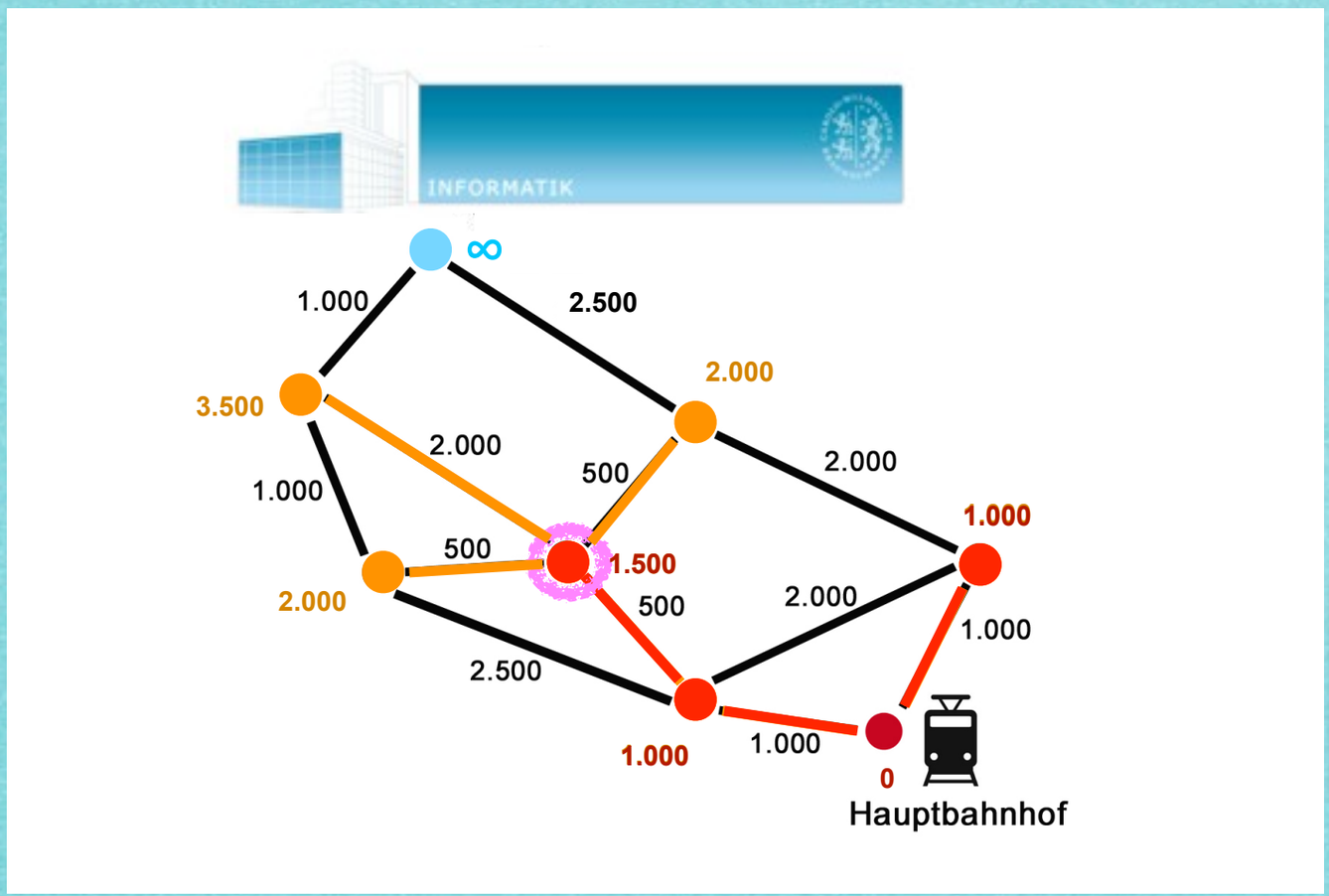
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

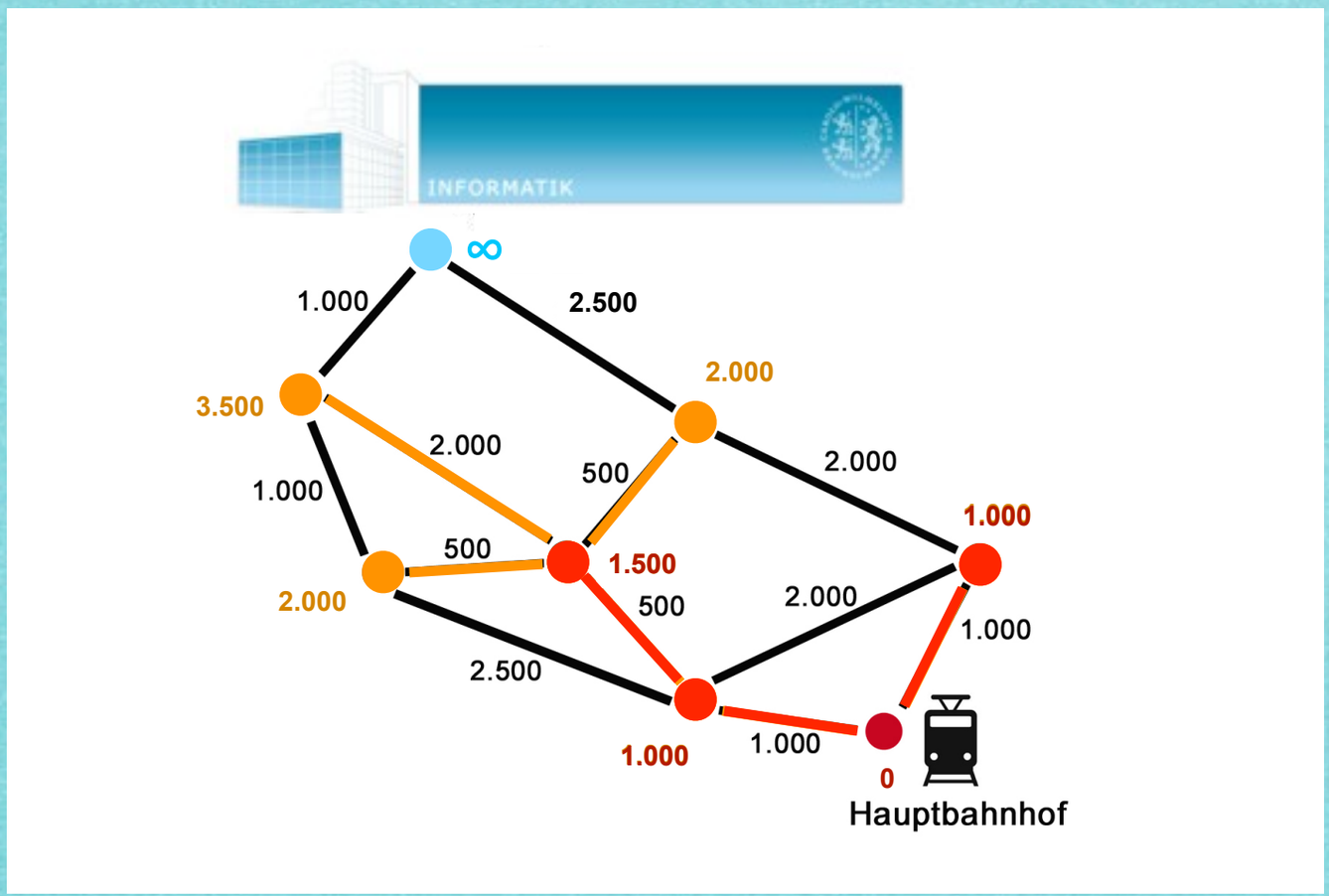
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

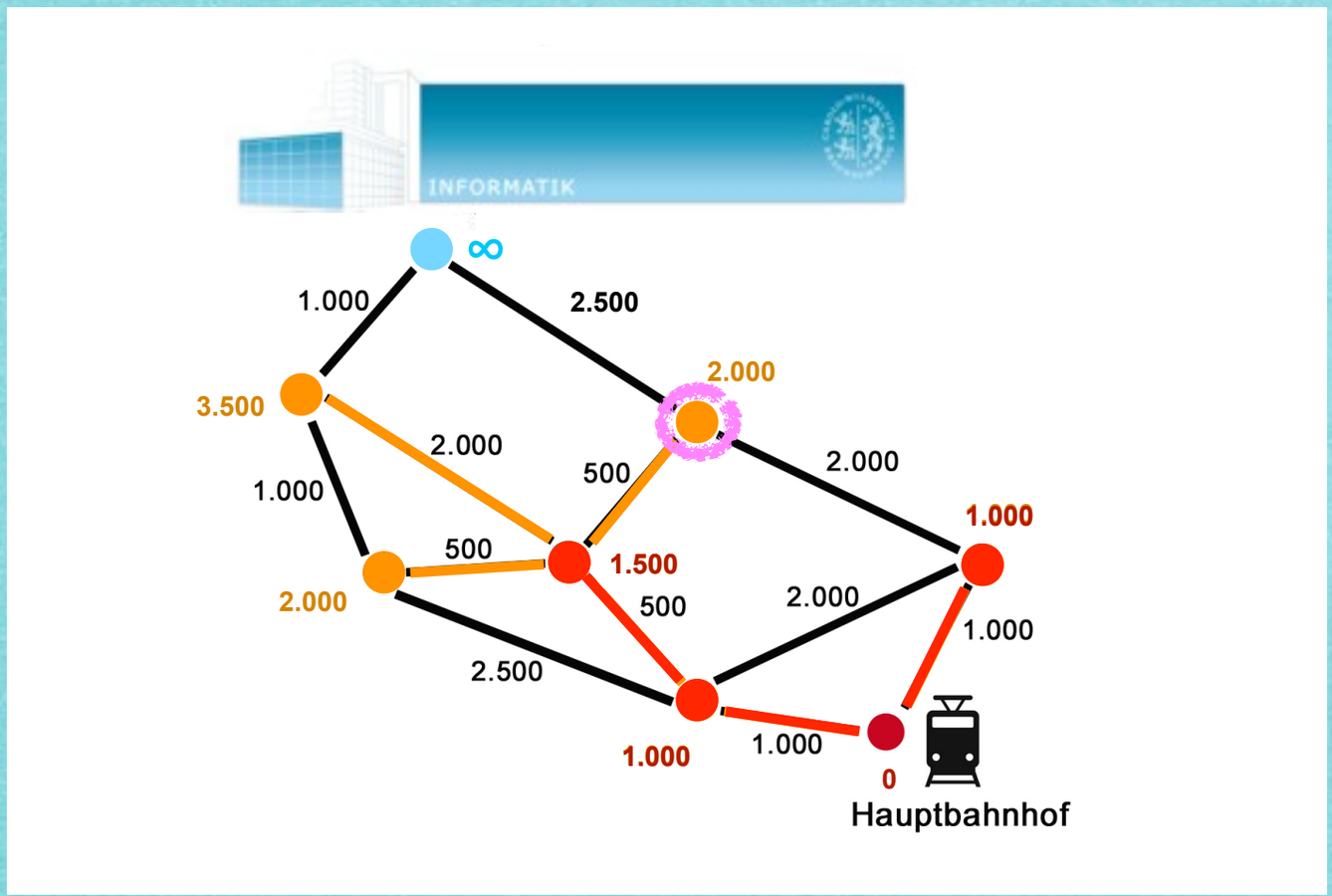
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

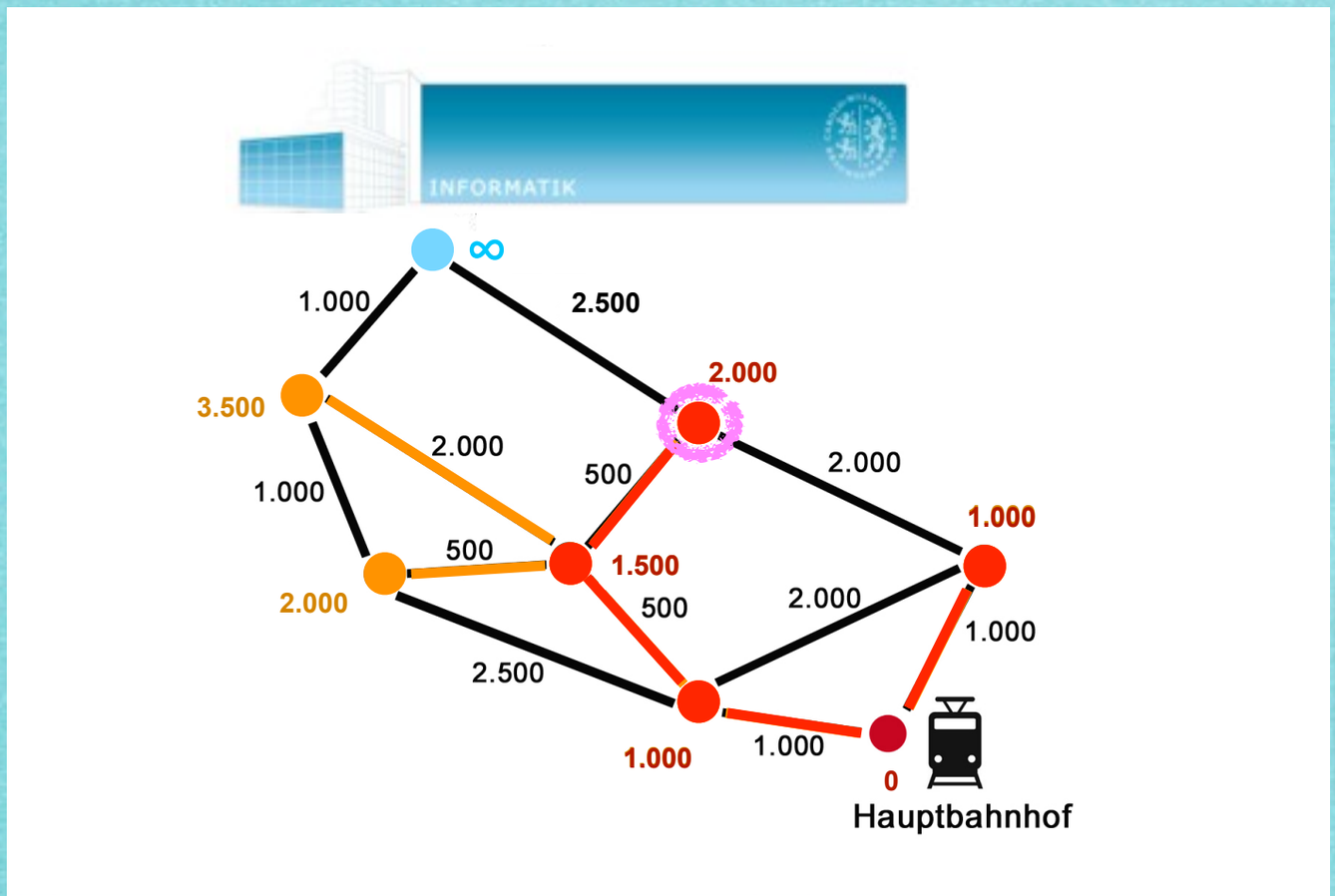
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

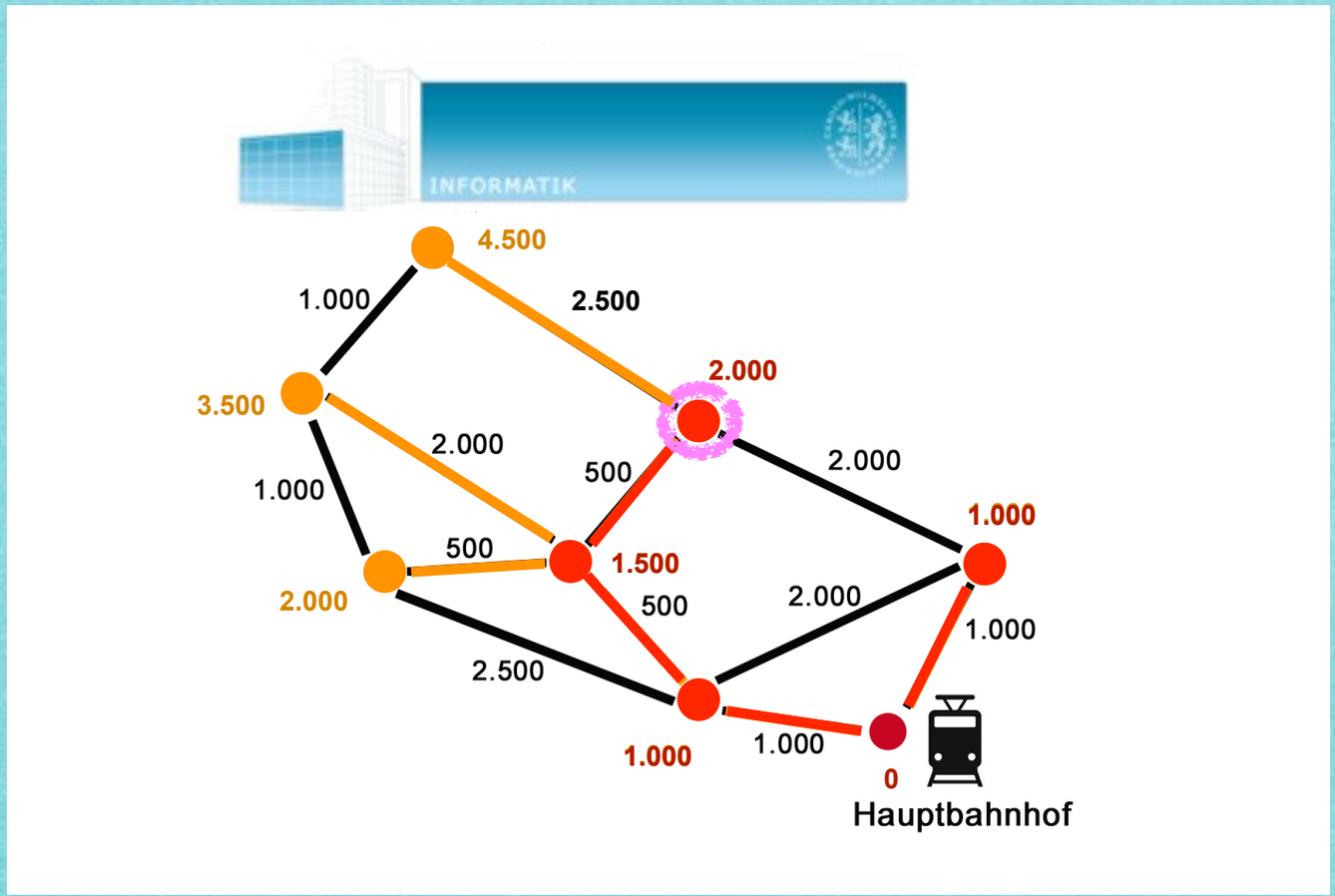
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

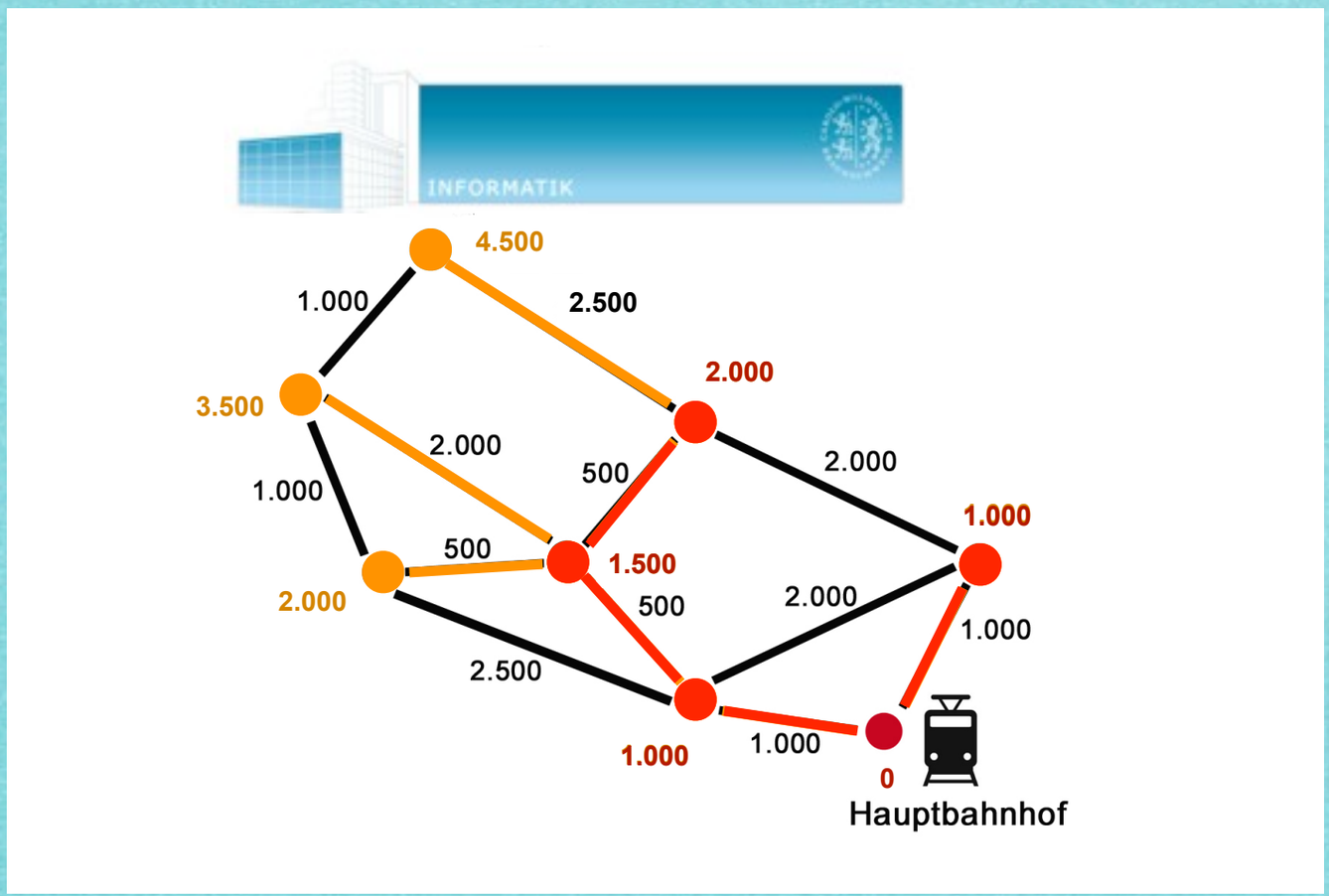
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

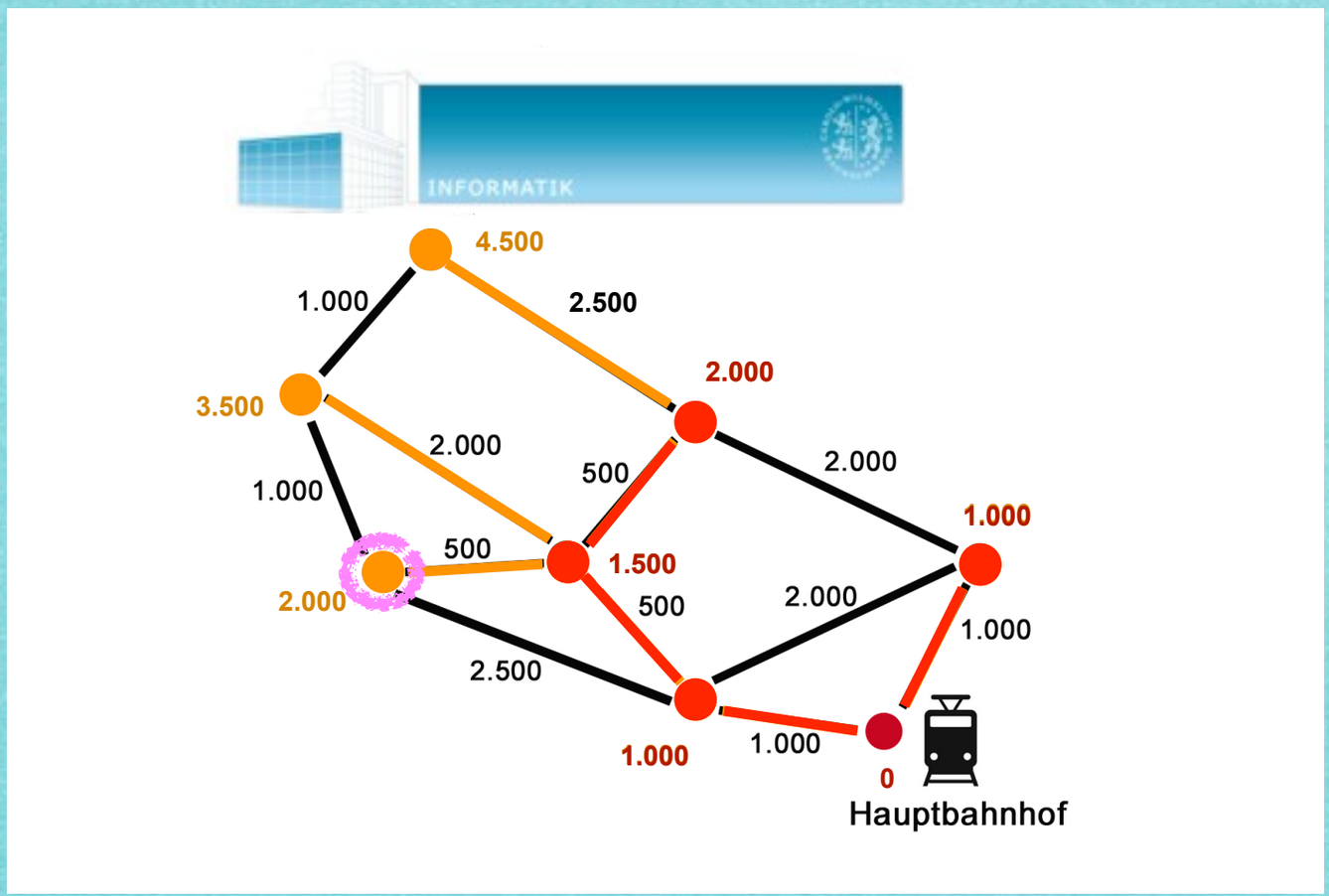
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

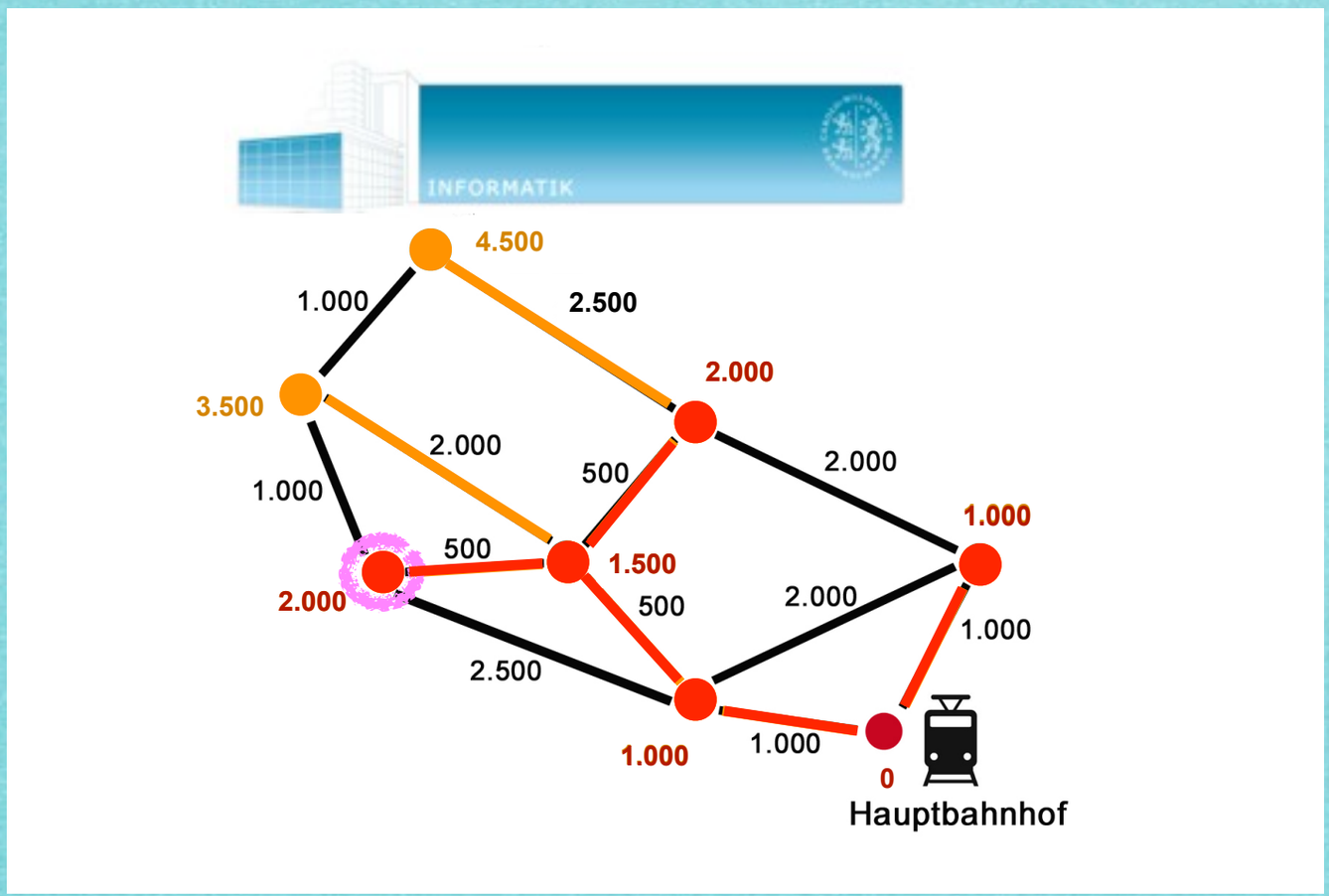
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

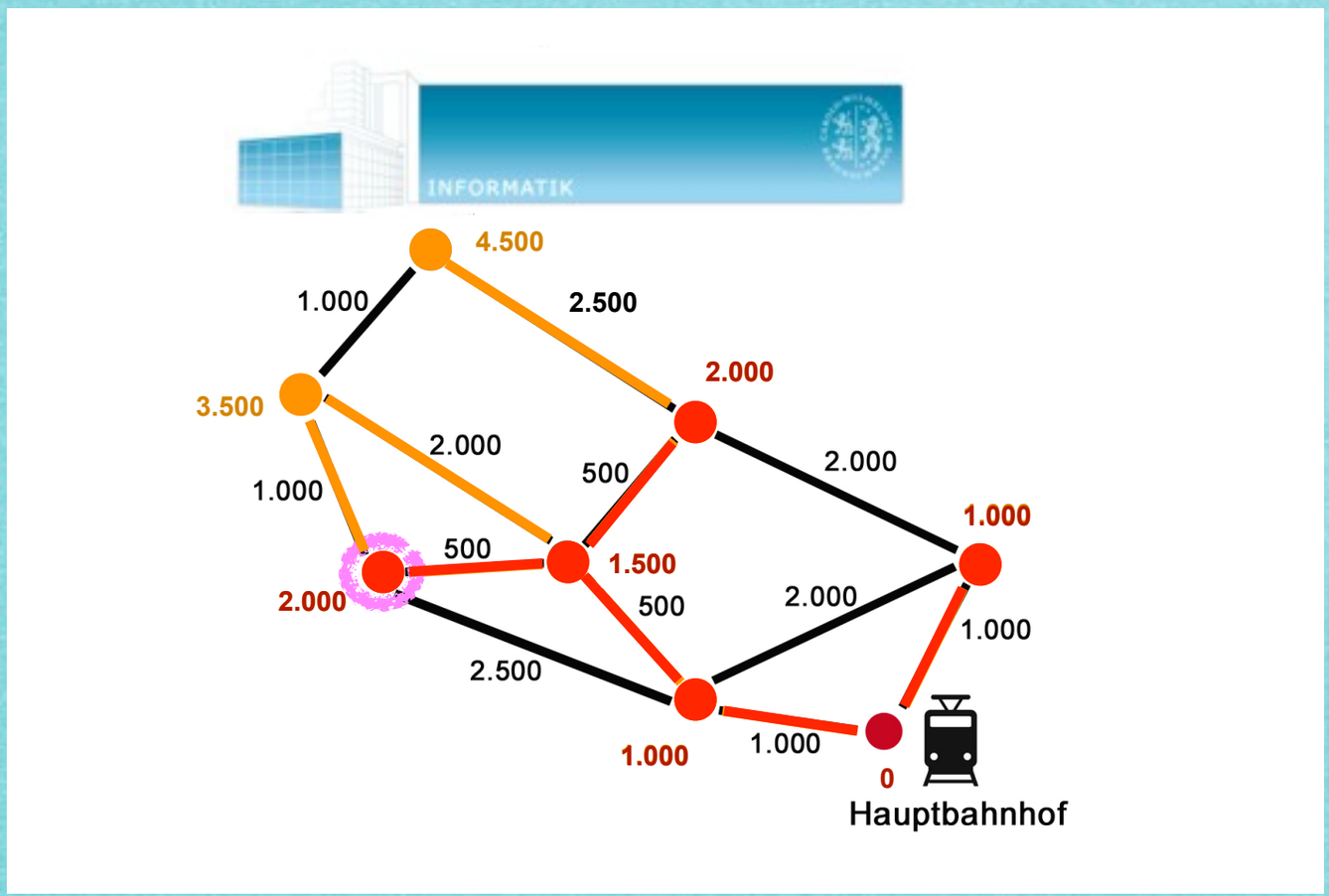
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

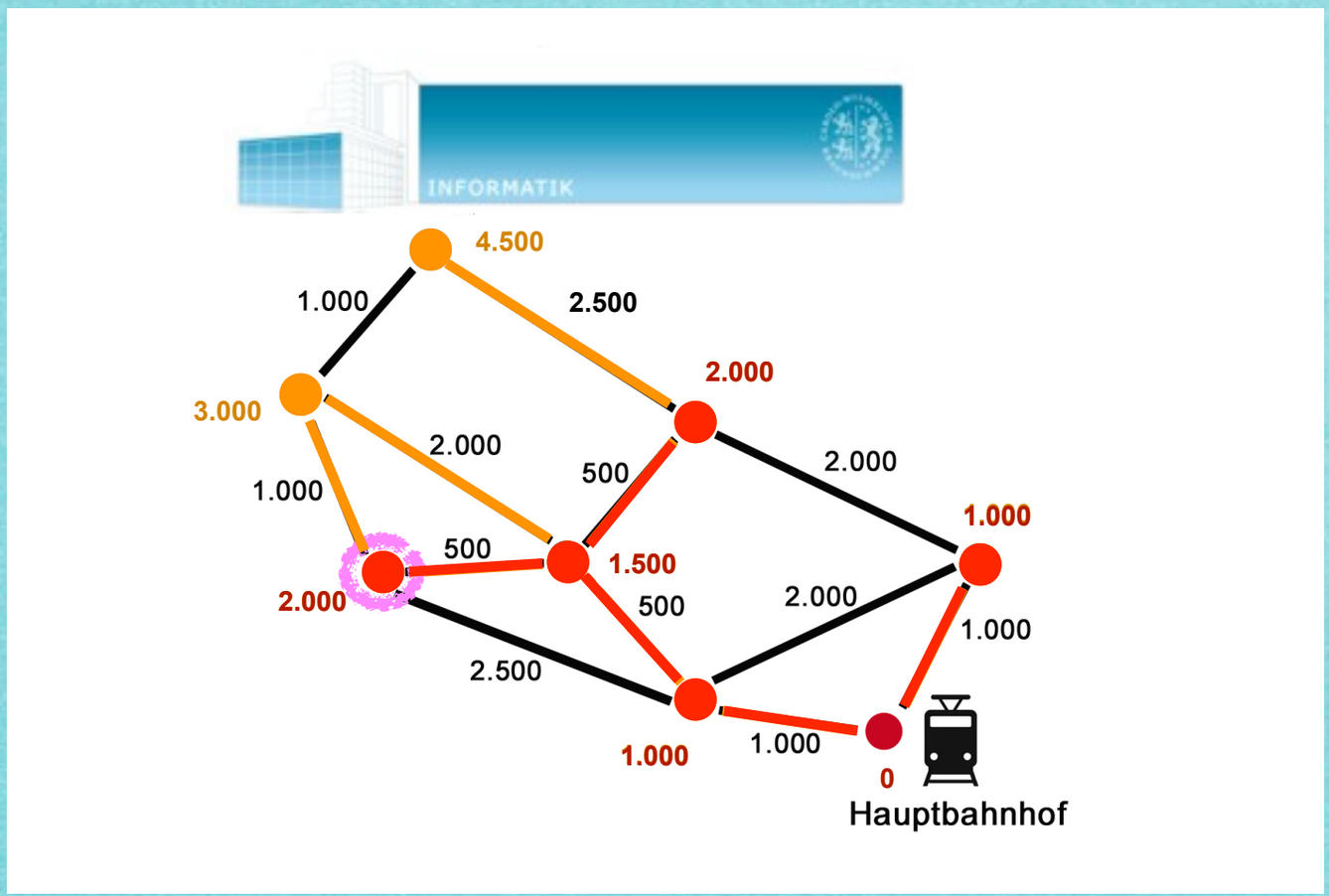
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

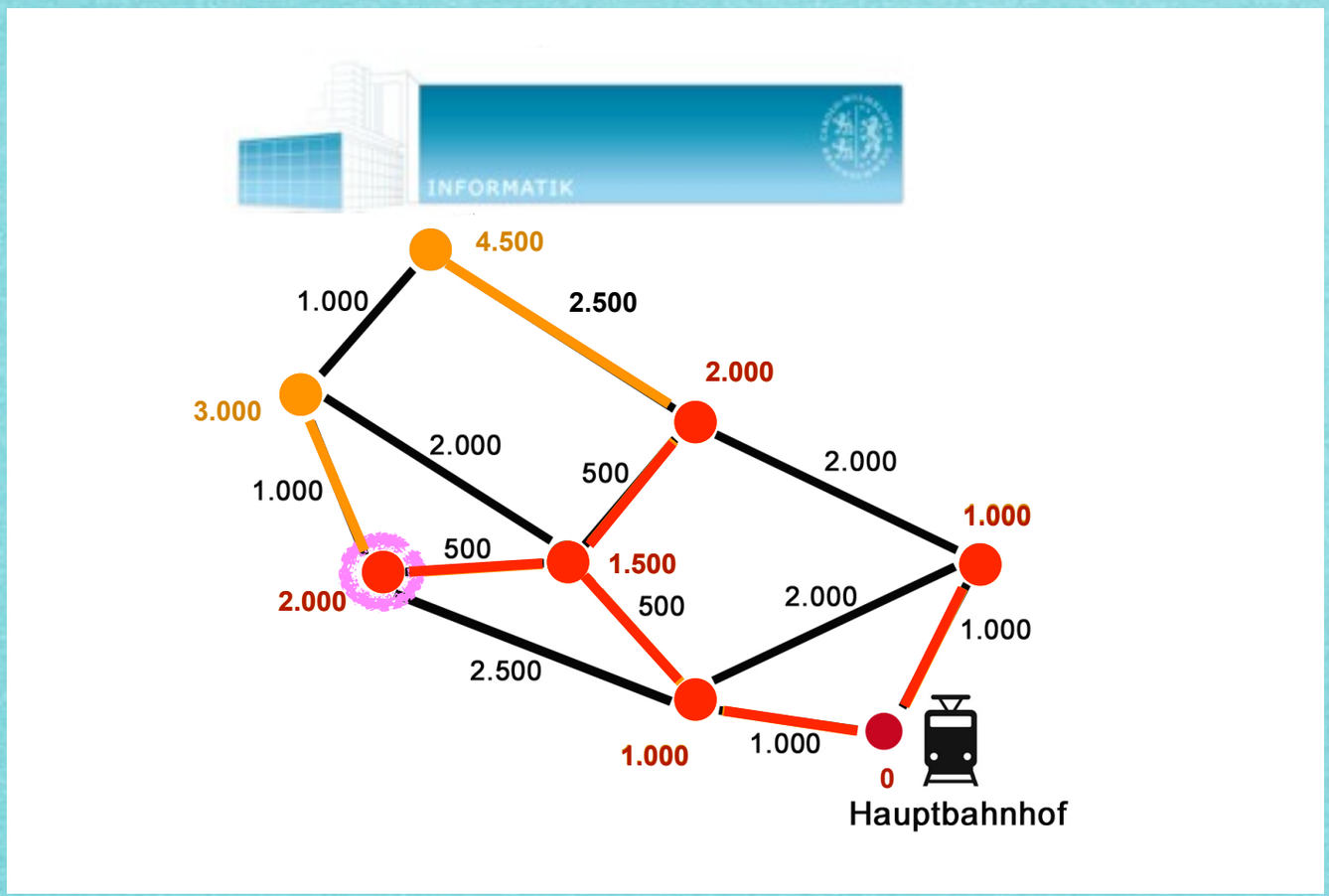
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

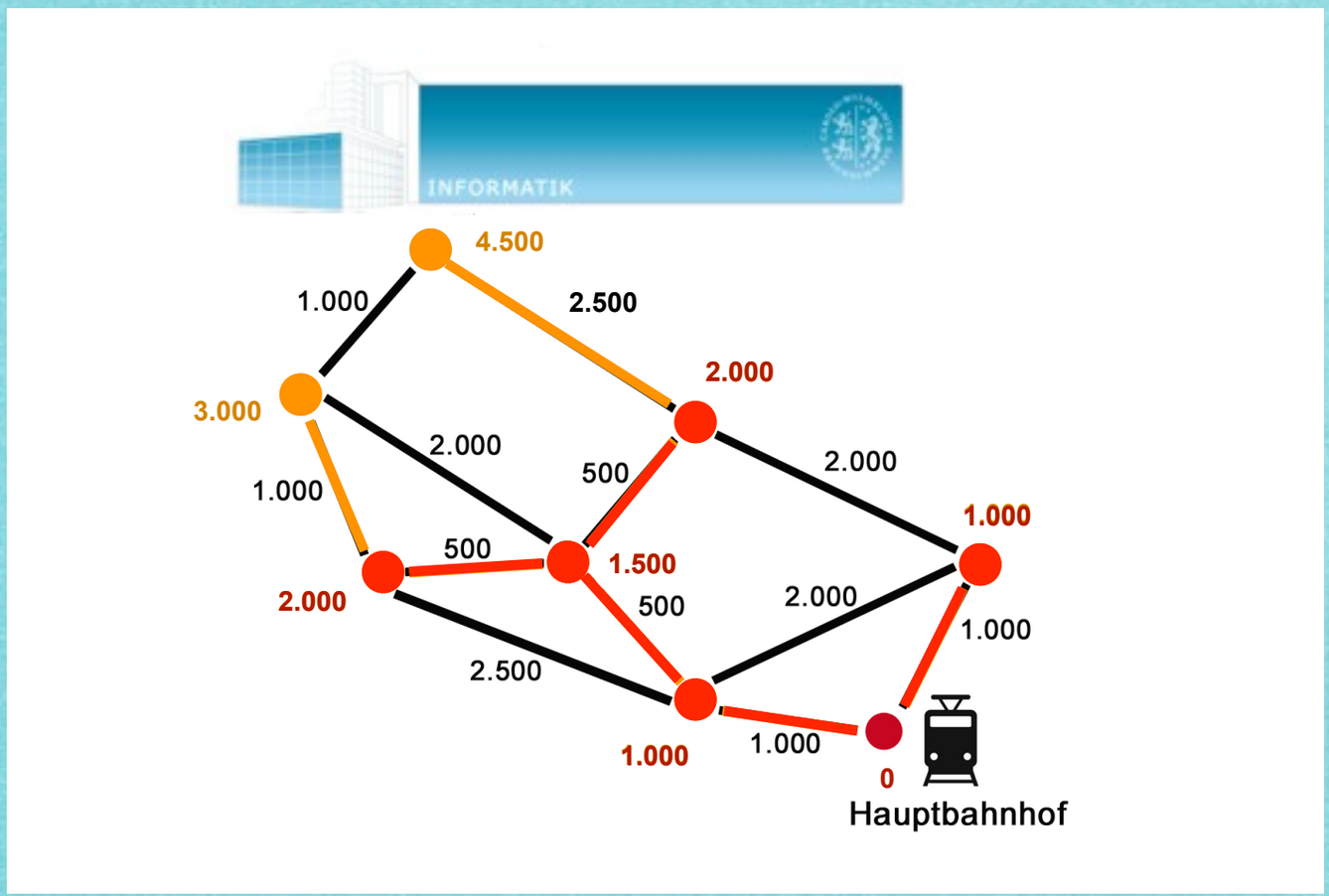
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

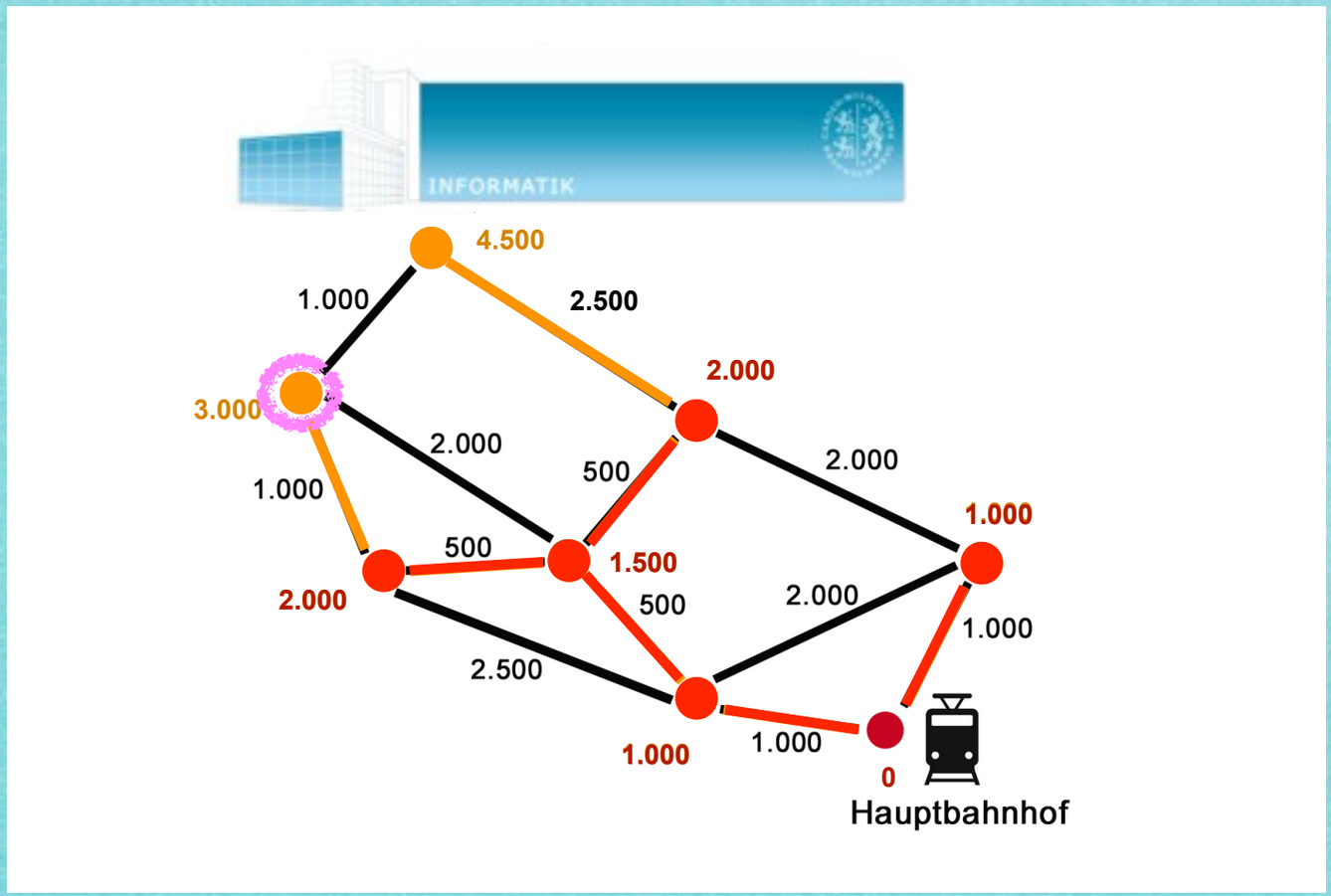
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

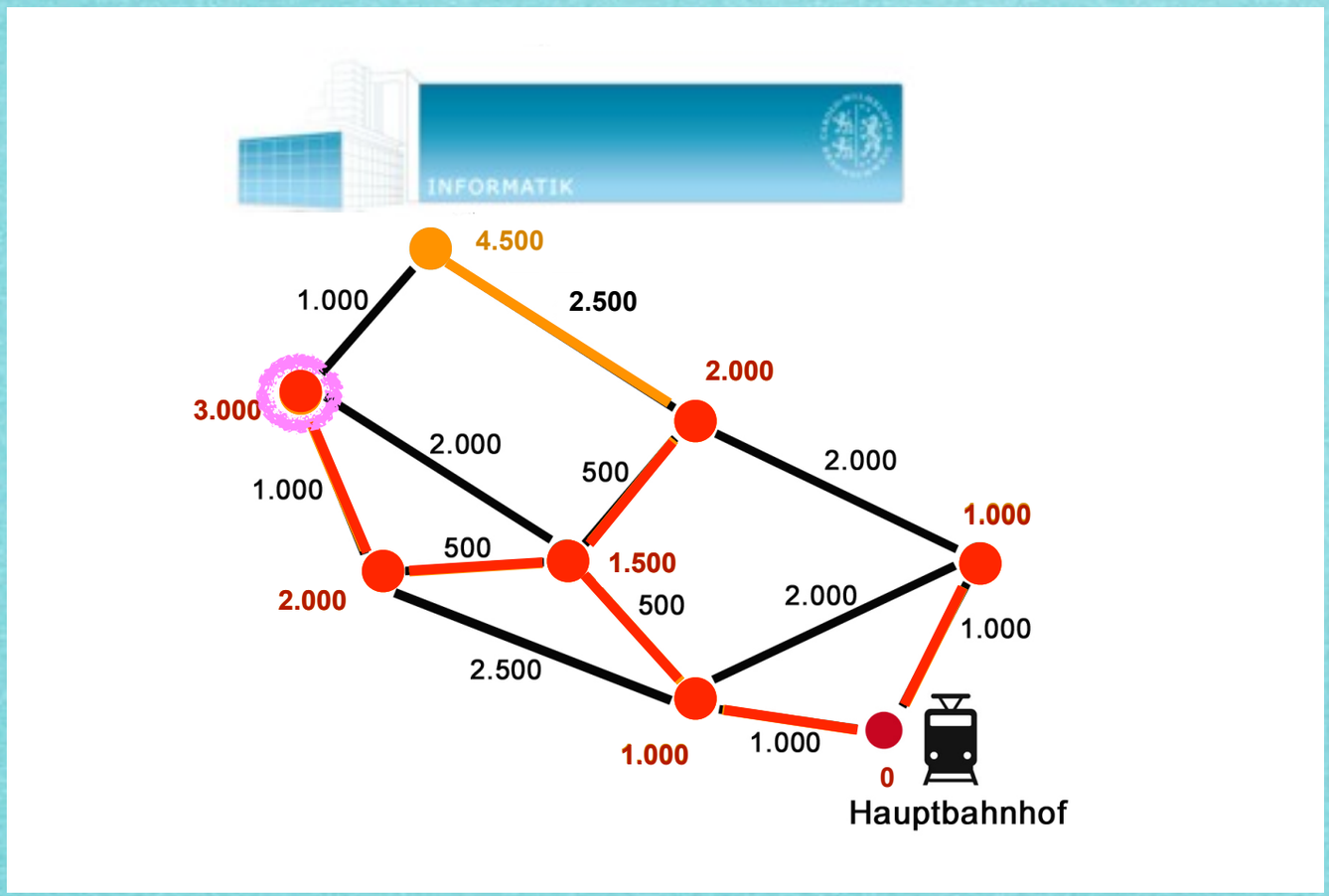
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

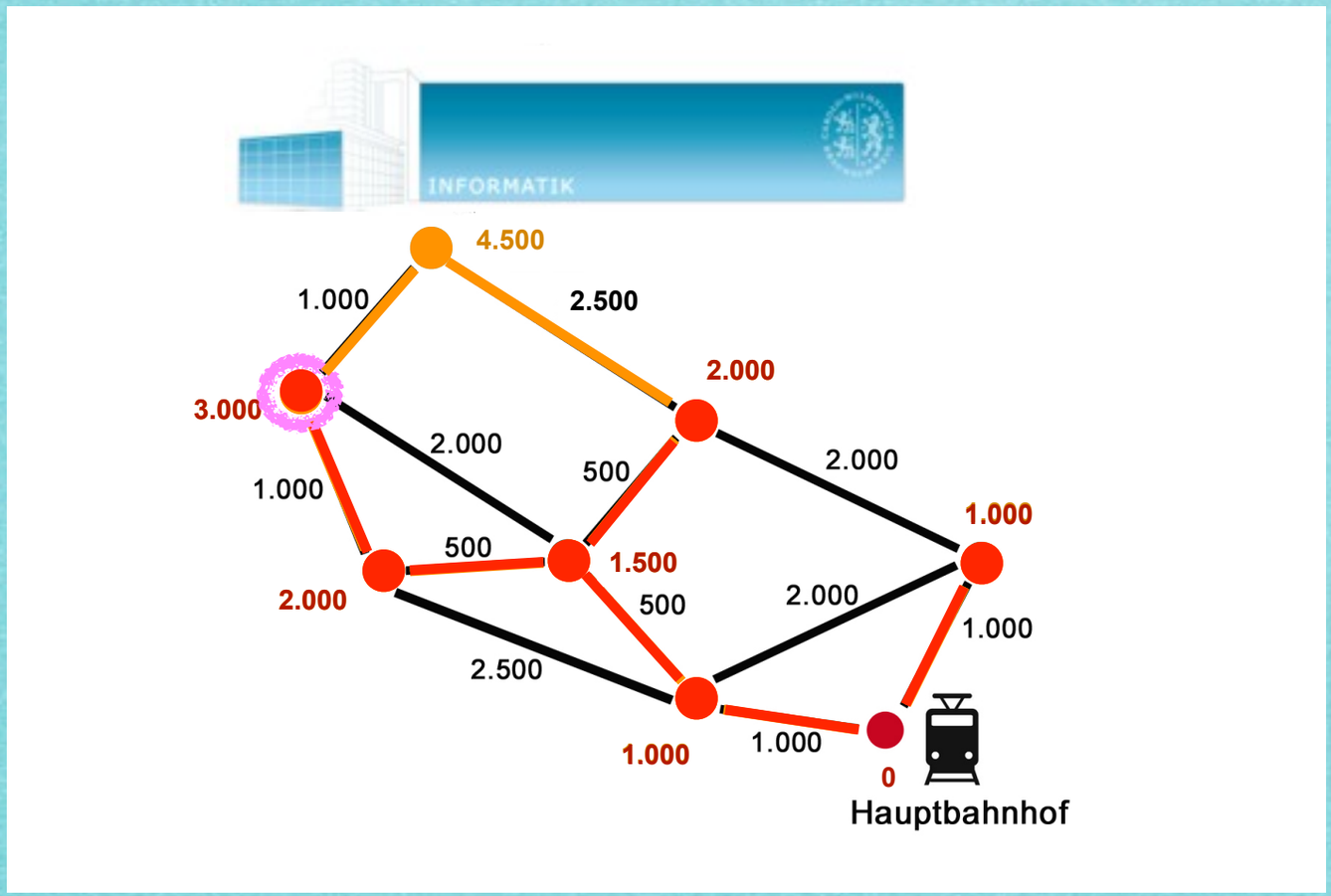
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

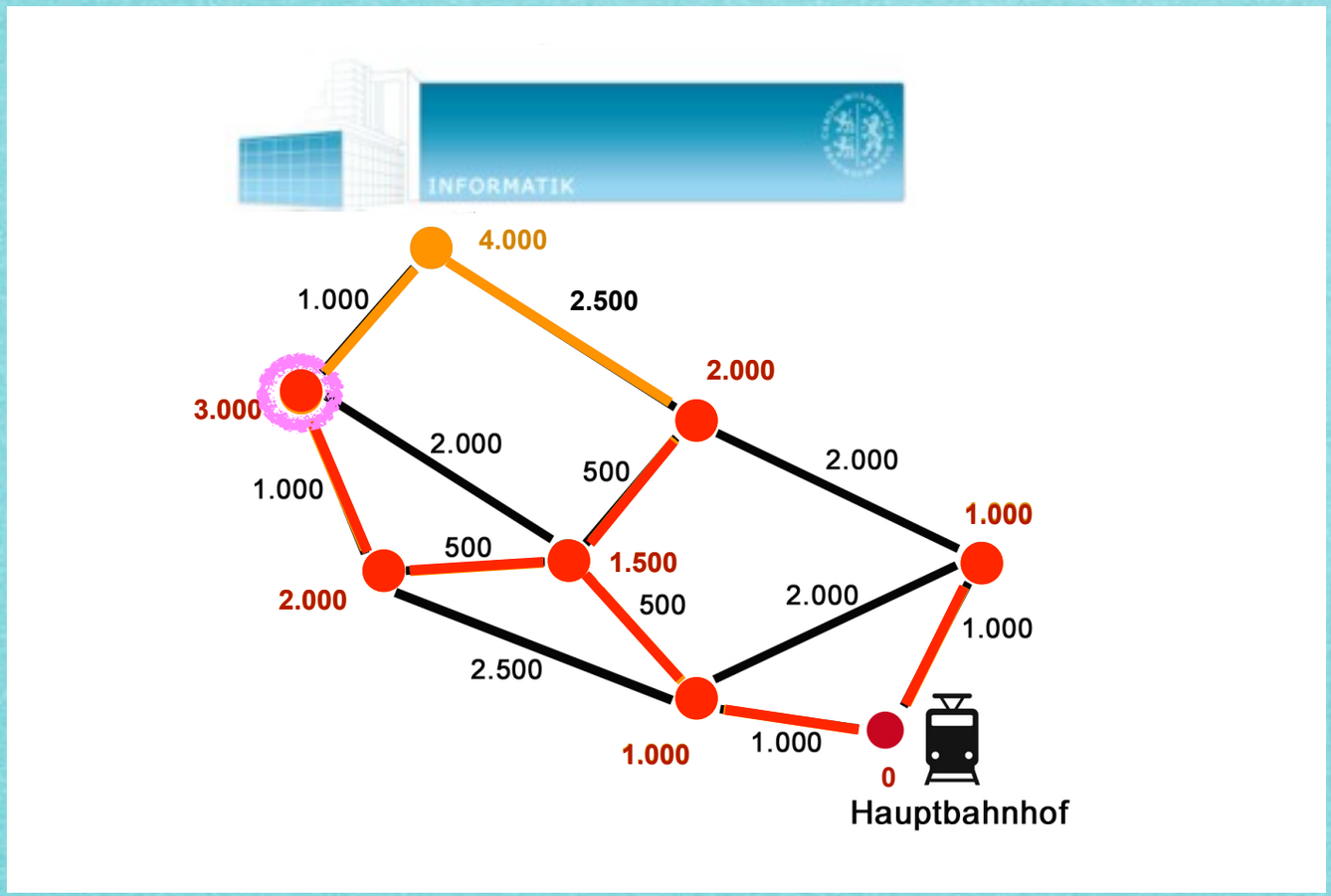
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

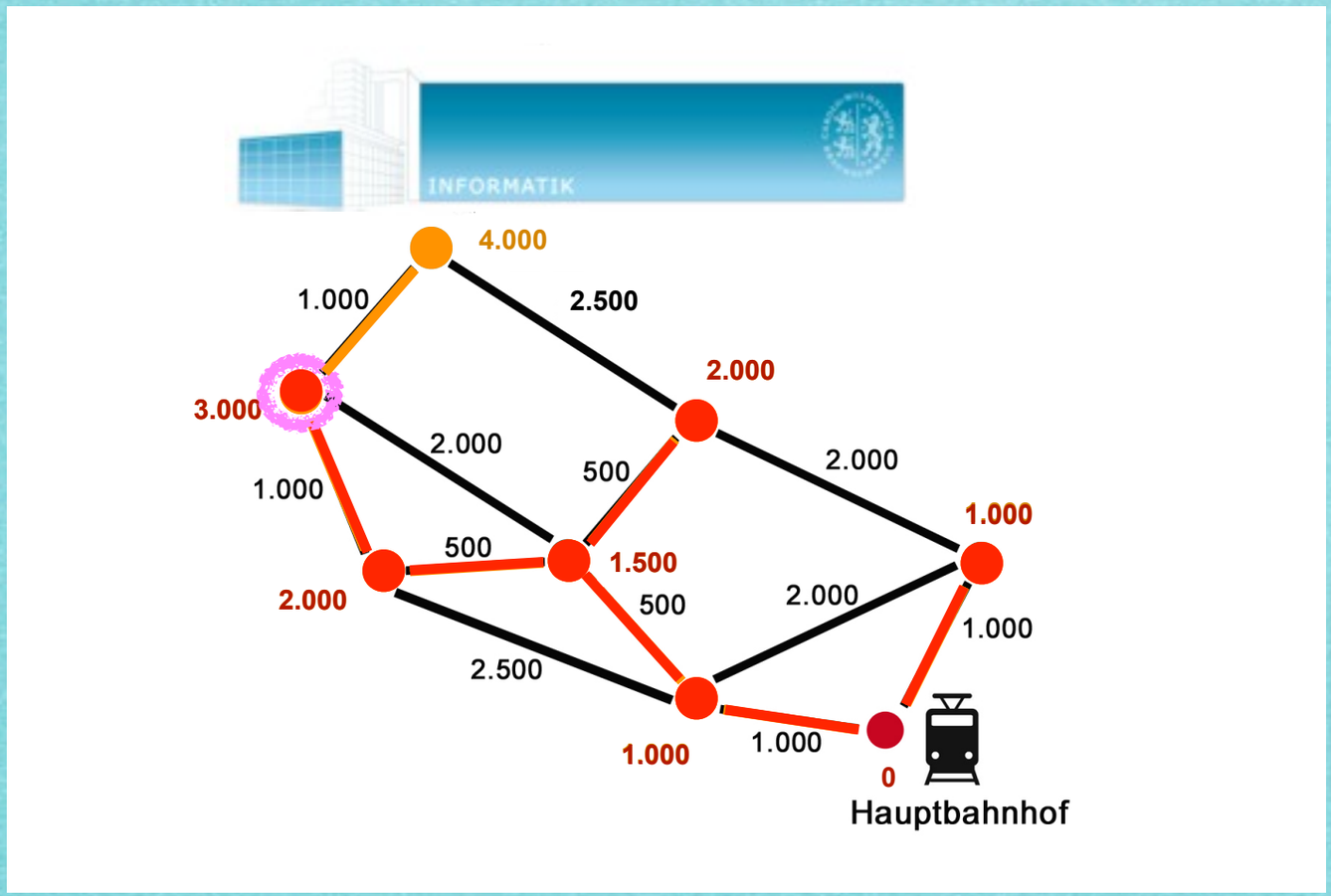
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

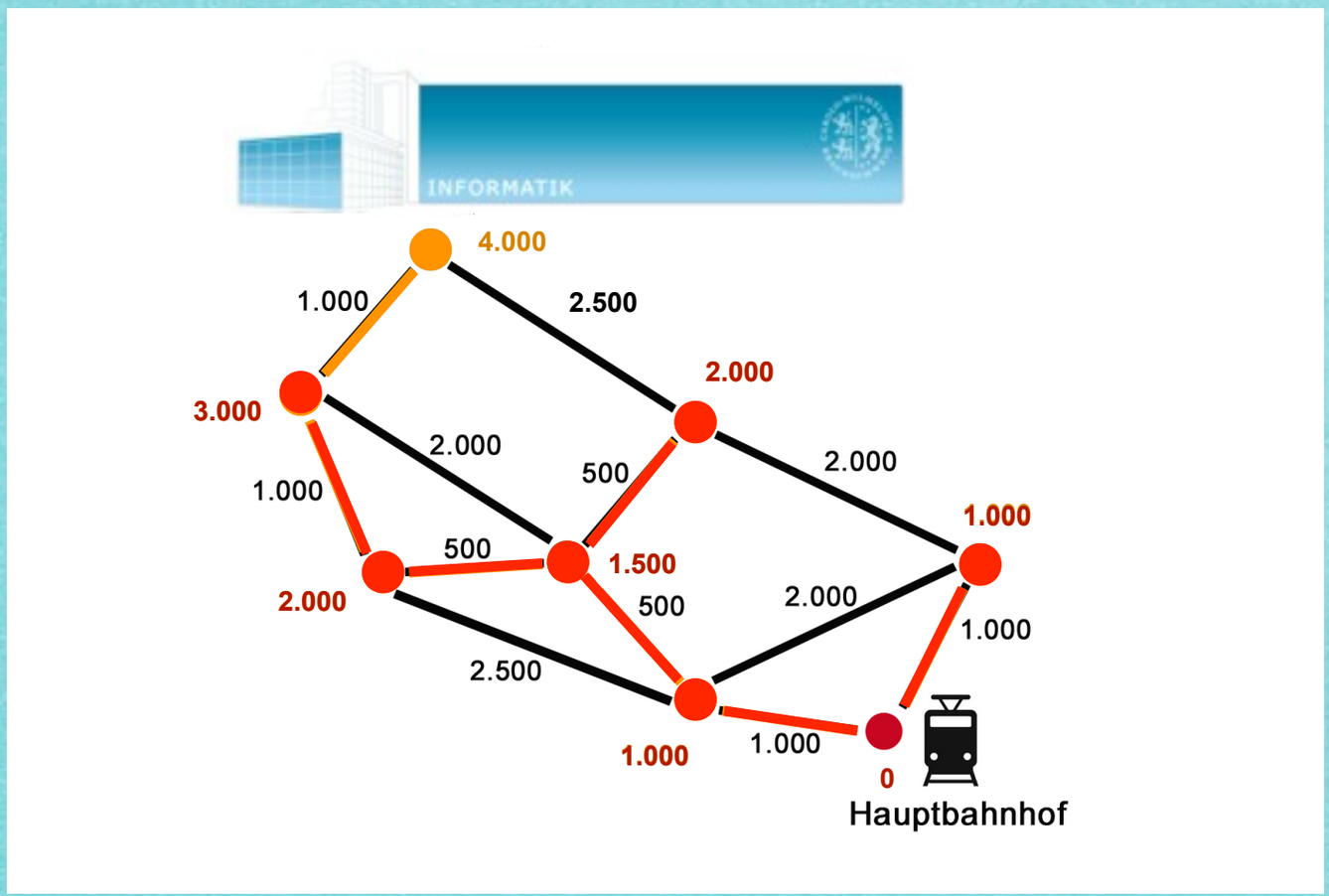
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

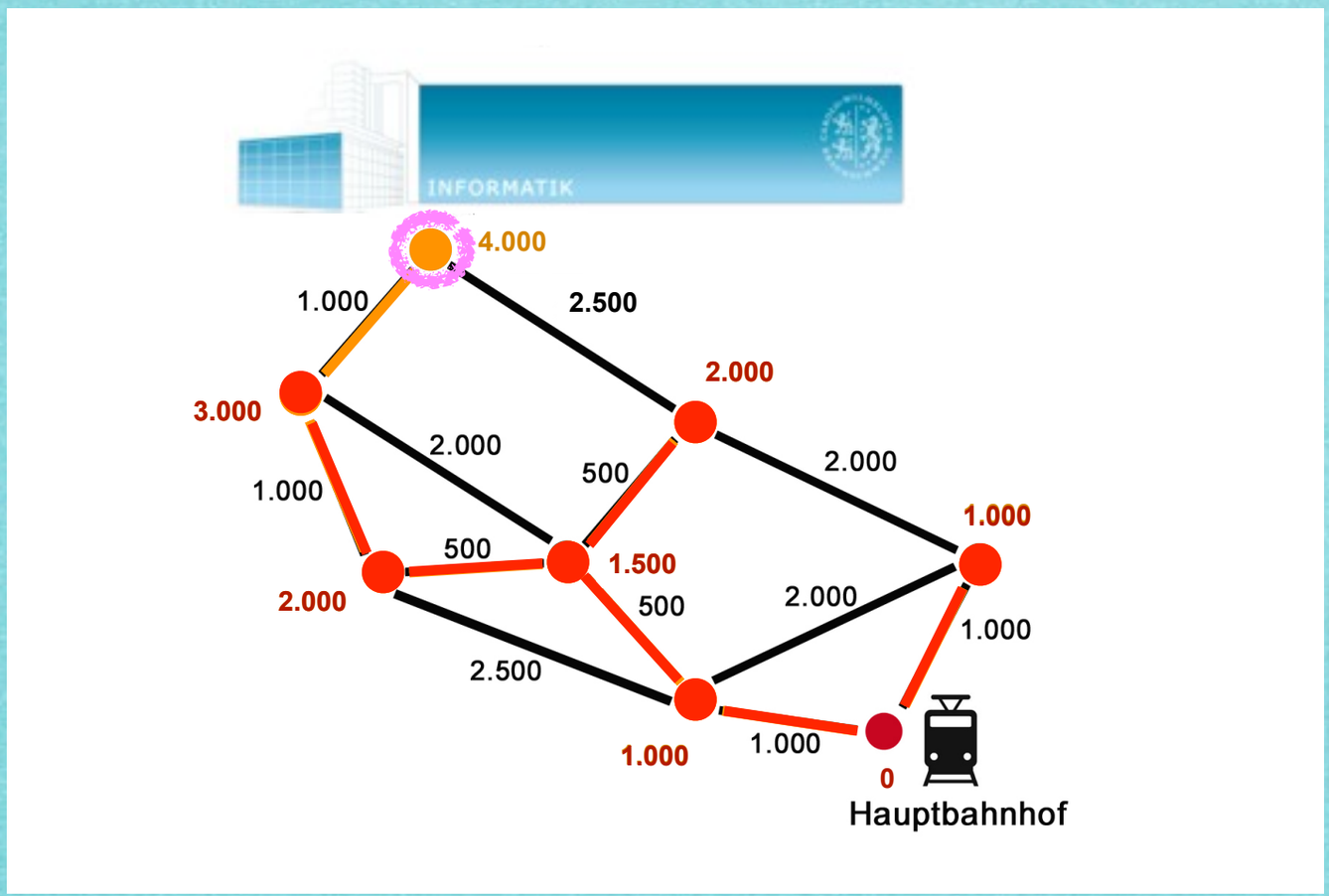
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

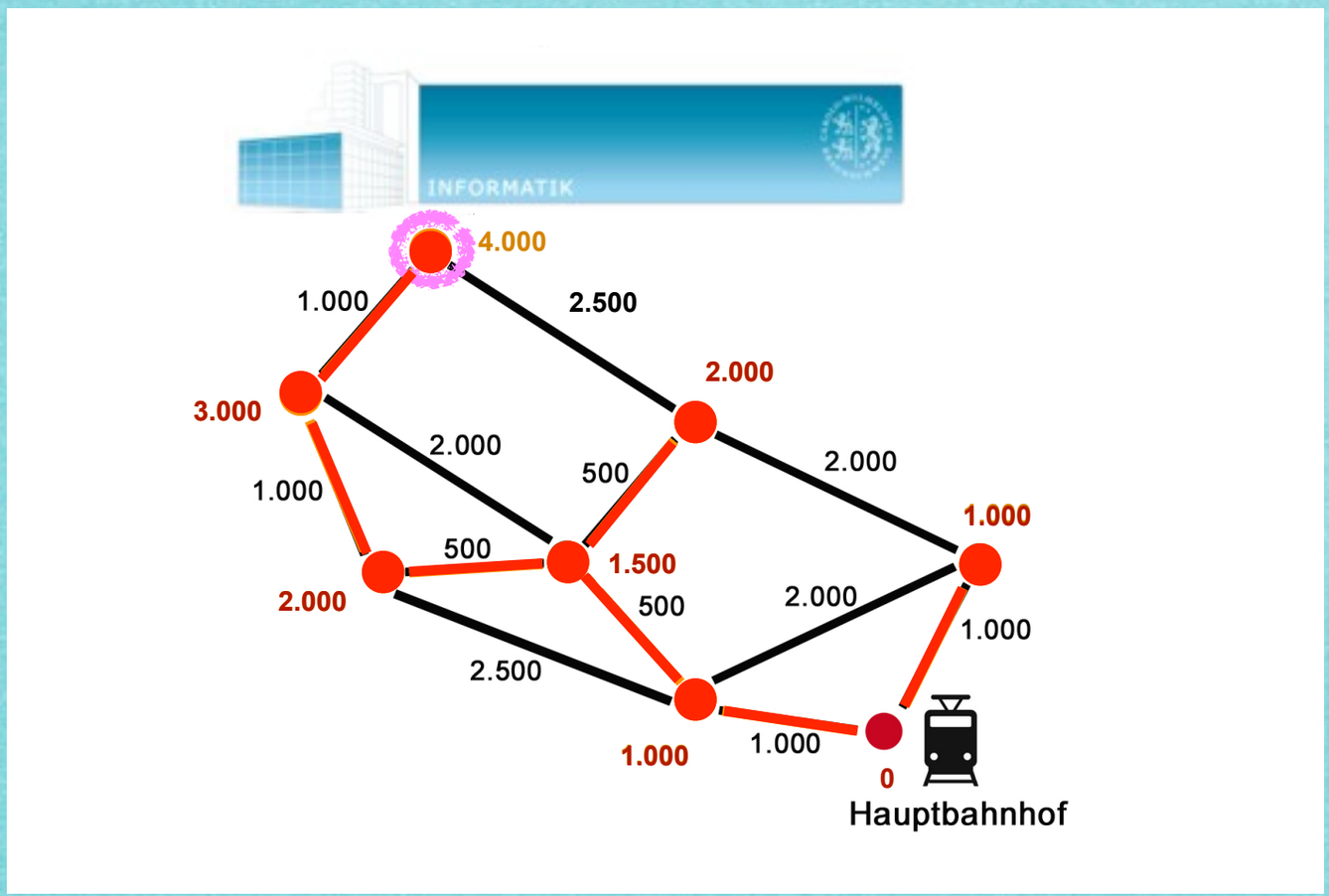
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

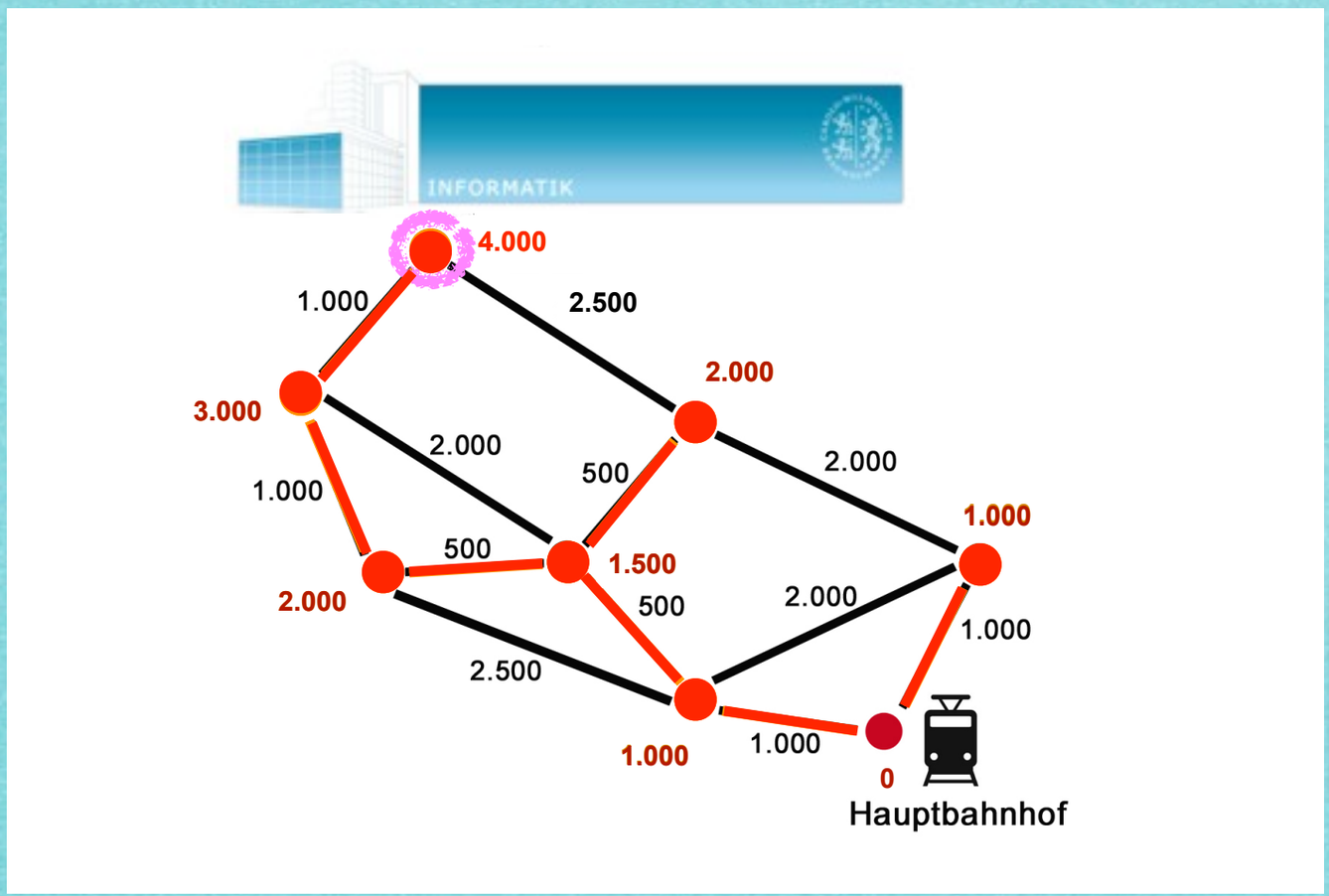
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

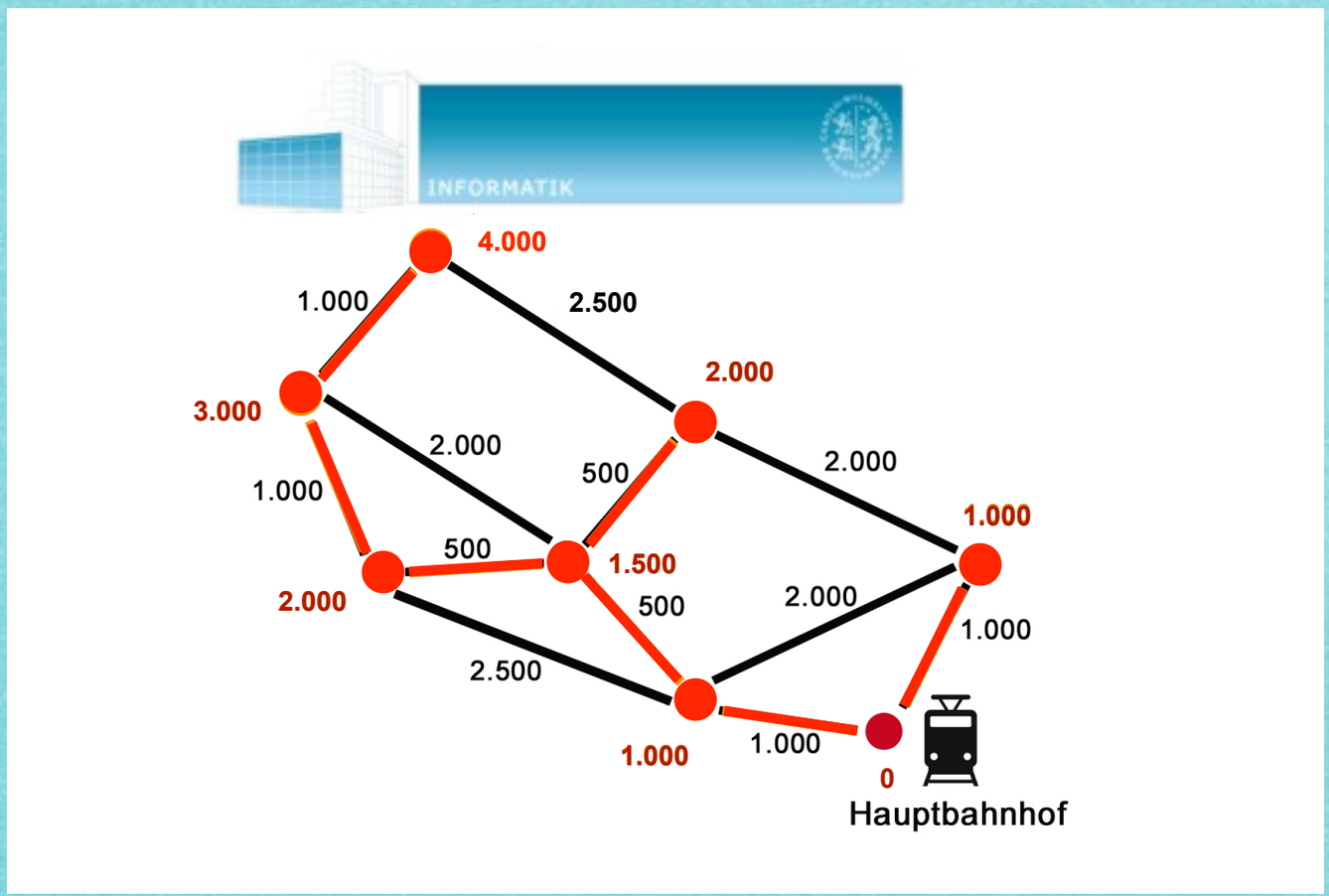
Dijkstras Algorithmus



Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

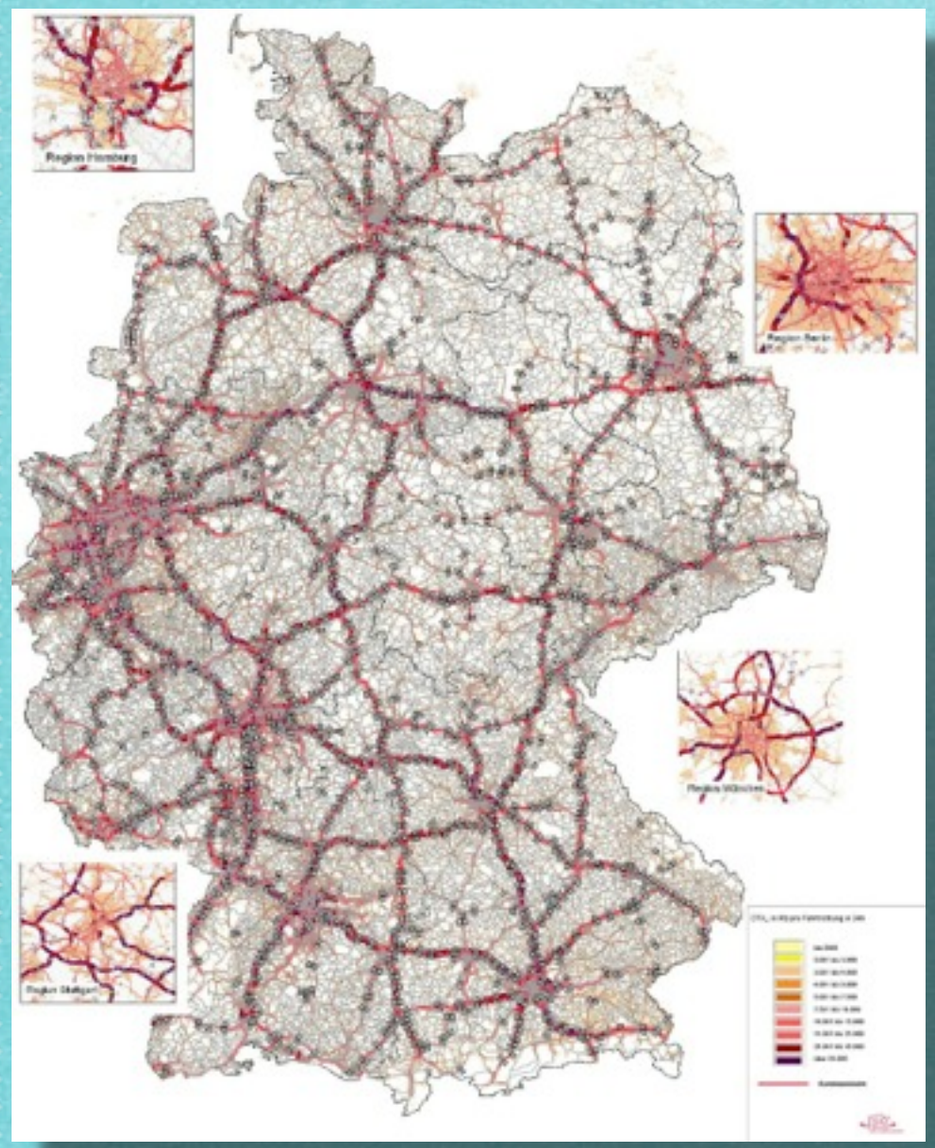
Dijkstras Algorithmus

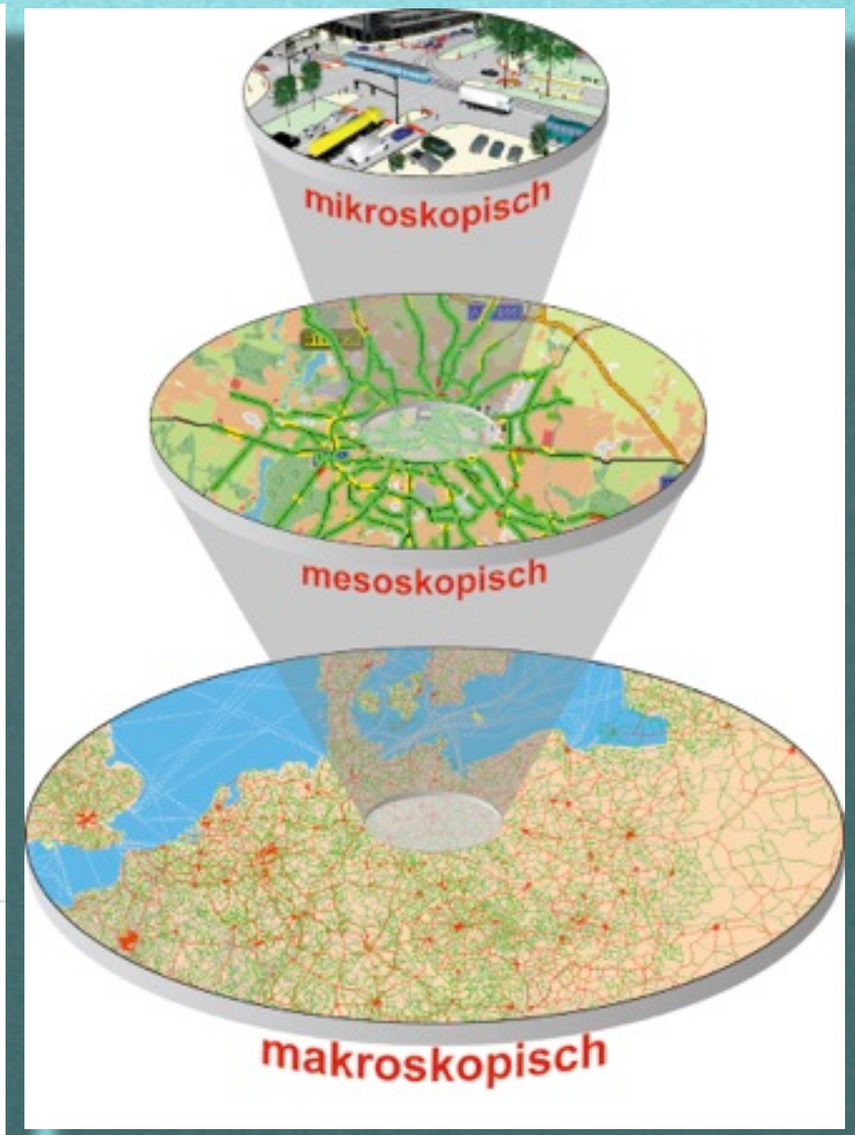


Edger Dijkstra
(1930-2002)

Funktioniert auch für verschieden lange Kanten

Routenplanung und Verkehrsplanung





Andere Verkehrsoptimierung

Wissen aus Braunschweig

Forscher der TU um Professor Sándor Fekete wollen Autofahrer mit besseren Daten über den vorausfahrenden Verkehr versorgen. Mancher Stau könnte sich dadurch schneller auflösen. Technisch ist diese Kommunikation kein Problem.

Der Verkehr könnte besser fließen

Die Autoschlange, durch Kommunikation verbunden – TU-Arbeitsgruppe stellt neues Informationssystem vor

Von Harald Duin

Ein glücklicher Tag. Ich erwische den Stauforscher Sándor Fekete kurz vor seinem Flug nach Singapur in seinem Büro. Logisch, in Singapur gibt es sicher jede Menge Staus. Doch dorthin eilt er aus einem anderen Grunde. In Singapur findet der Weltkongress für Origami statt, landläufig die Kunst des Papierfaltens.

Doch mit Origami kann man nicht nur Kraniche und andere Flugobjekte knicken. Mit dieser Technik lassen sich auch Anwendungen in Medizin, Technik oder Raumfahrt optimieren. Fekete nennt als Beispiel das Falten und Entfalten von Airbags. Damit keine Missverständnisse aufkommen: Er persönlich wird in Singapur keine Airbags falten. Wohl aber kann man sagen, dass seine Tätigkeit im Institut für Betriebssysteme und Rechnerverbund ihm alle Chancen zur Entfaltung bietet.

Zurück zum Stau. Fekete: „Gucken Sie sich doch mal an, was die Autos heute so alles können, aber untereinander sind sie völlig unvollkommen vernetzt." Schön wäre es zum Beispiel beim Stop-and-Go-Verkehr, mehr zu wissen, was die

Alles fließt? Schön wär's. Der Stau ist eine fast tägliche Erfahrung. Wie hier auf der A2. Foto: Archiv

Andere Verkehrsoptimierung

Verkehr, mehr zu wissen, was die anderen Autos davor tun.

Manche Staus scheinen wie aus einem Nichts zu entstehen. Man spricht auch von einem Phantomstau. Der Vorgang ist freilich seit Jahren enträtselt. Fekete: „Bei hohen Verkehrsdichten kann sich schon eine kleine Störung, etwa ein starkes Abbremsen oder ein zu dichtes Auffahren, nach hinten verstärken, durch den gesamten Verkehr fortpflanzen und ein ‚Schwingen' des Verkehrssystems auslösen."

Wenn sie einmal da sind, lassen sich die Stauwellen nicht mehr so einfach ausbügeln. Schon kleine Störungen genügen, um die Wellen zu erzeugen. Wie so ein „Ruckeln" entsteht, lässt sich sowohl in Simulationen wie auch in Experimenten beobachten. Fekete: „Das Auftreten von Stauwellen ist in den physikalischen Gesetzen verankert."

„Das Auftreten von Stauwellen ist in den physikalischen Gesetzen verankert."

Sándor Fekete, Verkehrsforscher

Was bringt der Versuch der TU-Forscher, Fahrzeuge mit besseren Daten über den vorausfahrenden Verkehr zu versehen? Eine ganze Menge. Bereits jetzt kann man mit drahtloser Kommunikation Informationen über Ort und Geschwindigkeit von Fahrzeugen weitergeben, und es existieren, wie Fekete erläutert, auch schon Systeme, mit denen sich ein Auto an die Geschwindigkeit eines unmittelbar vorausfahrenden Autos anpassen kann.

Die beschriebenen Stauwellen lassen sich jedoch damit nicht beseitigen. Hierfür muss die Wechselwirkung zwischen den Fahrzeugen so beeinflusst werden, dass sich auch die physikalischen Gesetzmäßigkeiten für den Fahrzeugfluss verändern.

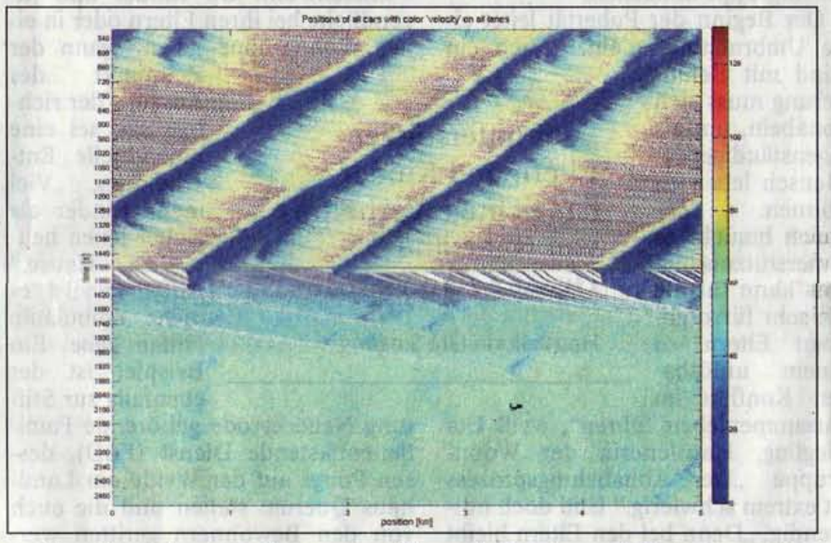
Die Arbeitsgruppe um Sándor Fekete hat nun ein einfaches Regelwerk entwickelt, mit dem Autos die Informationen mehrerer vorausfahrender Autos auswerten können, um den Verkehrsfluss im Stau zu glätten. Erkennt der Fahrer, gut informiert, dass sich der vorausfahrende Verkehr verlangsamt, muss er nicht unnötig beschleunigen.

Das hat einen verblüffenden Effekt auf den gesamten Verkehrsfluss.

Fekete: „In Simulationen zeigt sich eine Treibstoffersparnis bis zu 40 Prozent." Der Verkehr kann gleichmäßiger fließen, und der Stau kann sich sogar wieder auflösen, selbst wenn nur ein Teil der Fahrzeuge mitspielt.

Da jeder Fahrer Sprit spart, gibt es einen starken Anreiz, das System zu benutzen und die Regeln einzuhalten. Und die Nerven werden geschont. Freilich muss das ganze an der TU ersonnene System tatsächlich noch eingebaut werden. Das ginge auch bei älteren Autos. Und die Industrie muss mitspielen. Man darf sich freilich schon ein bisschen wundern, dass die großen Autokonzerne dieses Thema bisher wenig beachteten.

Der Stau und die Intelligenz, seine Schockwellen zu mäßigen. Ein Graffito an der A 40 verweist auf den eigentlichen Verursacher: „Ihr steht nicht im Stau, Ihr seid der Stau!"



Grafik „Simulation von Stau und Auflösung": Im oberen Teil der Simulation erkennt man typische Stauwellen, dargestellt als blaue Streifen. In der Bildmitte wird das neue, an der TU entwickelte Regelwerk eingeschaltet. Der Stau löst sich weitgehend auf. Bild: Sándor Fekete

LEBENSDATEN

Professor Dr. Sándor Fekete

45 Jahre. Studium der Mathematik und Physik in Köln, Diplom 1989. Promotion (Mathematik) in Waterloo (Kanada) 1992. Forschungsaufenthalte in Stony Brook (USA), Tel Aviv, London, Newcastle (Australien), Kingston (Kanada) und am Massachusetts Institute of Technology (MIT).

Habilitation in Köln 1998. Danach an der TU Berlin. Seit 2001 Professor an der TU Braunschweig, zunächst in der Mathematik, seit 2007 in der Informatik.

Vielen Dank!

Vielen Dank!

Vielen Dank!

s.fekete@tu-bs.de