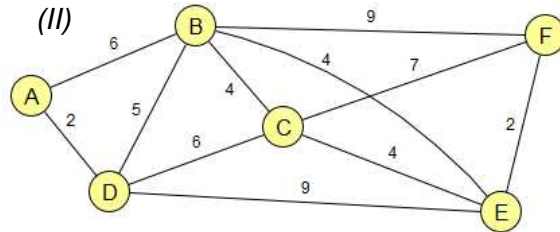
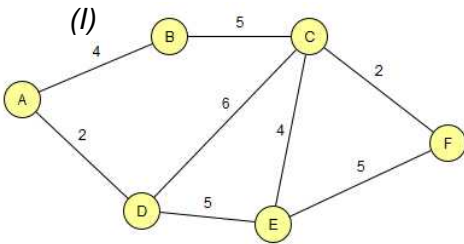


Das Problem des kürzesten Weges – Dijkstra und die Ameisen

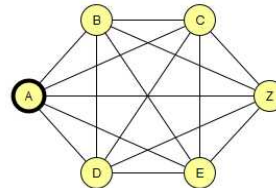
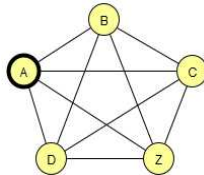
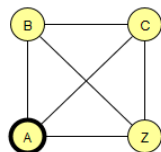
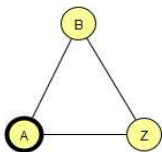
Wege in Graphen

**Aufgabe**

1. a) Welches ist der kürzeste Weg von **Alsdorf** nach **Fahrendorf**?
- b) Wie kannst du sicher sein, dass es keinen kürzeren Weg gibt?



2. Was wäre ein 'sicheres' Vorgehen, um zweifelsfrei den kürzesten Weg zu finden?
  3. Wie viele solcher Wege gibt es, die bei A beginnen, bei Z enden und jeden Knoten höchstens einmal besuchen? Wir gehen vom schlechtesten Fall aus, nämlich dass jeder Knoten mit jedem verbunden ist.
- (a) bei 3 Knoten   (b) bei 4 Knoten   (c) bei 5 Knoten   (d) bei 6 Knoten   (e) bei \_\_ Knoten

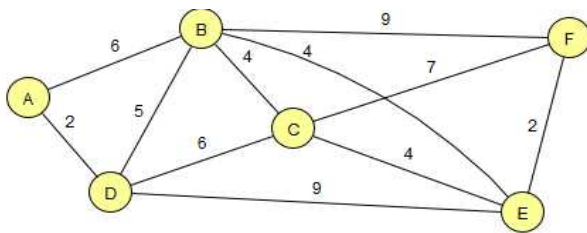


4. Ergänze die Ameisenregeln:

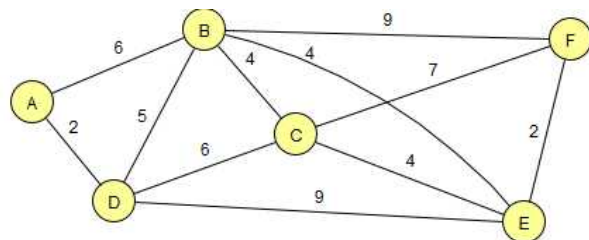
1. (Fast) unendlich viele Ameisen laufen am Startpunkt los.		
2. Alle Ameisen sind gleich schnell (1 Weeinheit in 1 Minute)		
3. Die Ameisen teilen sich an jedem Ort auf und laufen weiter.		
Zwei Ameisentrupps treffen unterwegs aufeinander, dann....	Zwei Ameisentrupps treffen ...	...

5. a) Simuliere wie die Ameisen laufen. Notiere jeweils den Zustand nach 6, 7, 9, ... Minuten.

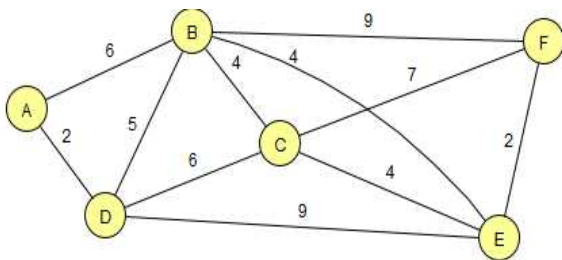
Nach 6 Minuten:



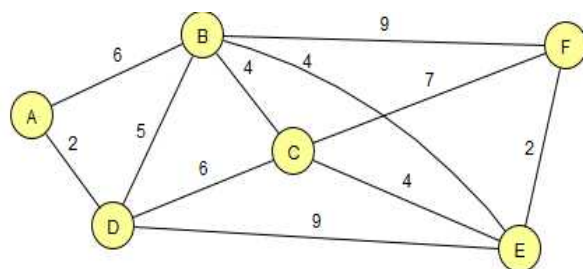
Nach (fast) 7 Minuten:



Nach 9 Minuten:



Angesommen nach \_\_\_ Minuten:



b) Welches ist der kürzeste Weg?

c) Wieso kannst du sicher sein, dass es keinen kürzeren Weg gibt?

d) Wenn es in einem Graphen zwei gleich lange, kürzeste Wege gibt, welchen würden die Ameisen finden? Überlege dir dazu ein Beispiel.

1 Ameisenbild: Quelle: Miriam Klein

**Der Dijkstra-Algorithmus**

**Edsger Wybe Dijkstra** \* 11. Mai 1930 in Rotterdam; † 6. August 2002 in Nuenen, war ein niederländischer Informatiker. Er war der Wegbereiter der strukturierten Programmierung. 1972 erhielt er den Turing Award für grundlegende Beiträge zur Entwicklung von Programmiersprachen. <sup>2</sup> (Bild <sup>3</sup>)



Schritt 1: Markiere den Startknoten als 'besucht' (rot). Beschrifte ihn mit der Distanz '0'. Er ist der aktuelle Knoten. 0 ●

Schritt 2: Für alle Nachbarknoten mache folgendes:

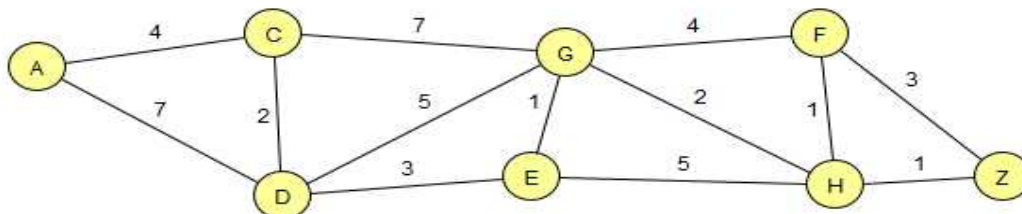
Berechne seine Entfernung zum Startknoten (= Entfernung am aktuellen Knoten + Entfernung zum Nachbarn)		
Falls der Nachbar noch keine Entfernung eingetragen hat,	Falls der Nachbar bereits eine Entfernung eingetragen hat, die größer ist als die eben berechnete,	Anderenfalls:
→ dann trage die gerade berechnete Entfernung ein und markiere den Weg:	→ dann ersetze sie durch die eben berechnete Entfernung. Lösche alle Wegmarkierungen zu diesem Knoten und markiere den gerade gefundenen Weg:	→ mache nichts.

Schritt 3: Falls es noch Knoten gibt, die nicht markiert (besucht) sind, wähle daraus denjenigen Knoten mit der kleinsten Entfernung und markiere ihn als 'besucht' (rot). Er ist der neue aktuelle Knoten. Mache bei Schritt 2 weiter.

Sind alle Knoten besucht (markiert) → Ende.

**Aufgabe:**

1. a) Ermittle mit dem Dijkstra-Algorithmus den kürzesten Weg von A nach Z.



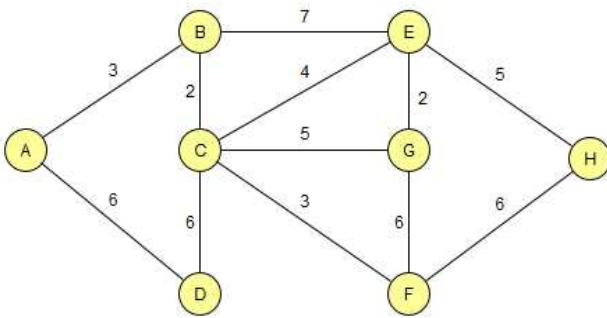
b) Welches sind die kürzesten Wege von A zu allen anderen Knoten?

<sup>2</sup> [https://de.wikipedia.org/wiki/Edsger\\_W.\\_Dijkstra](https://de.wikipedia.org/wiki/Edsger_W._Dijkstra) (abgerufen am 06.11.2018)

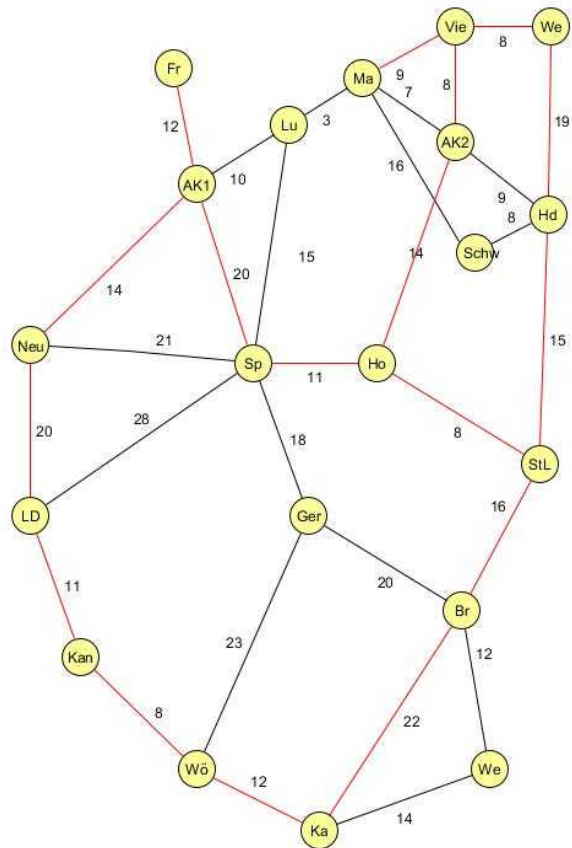
<sup>3</sup> [https://de.wikipedia.org/wiki/Edsger\\_W.\\_Dijkstra#/media/File:Edsger\\_Wybe\\_Dijkstra.jpg](https://de.wikipedia.org/wiki/Edsger_W._Dijkstra#/media/File:Edsger_Wybe_Dijkstra.jpg) (abgerufen am 06.11.2018) Hamilton Richards, manuscripts of Edsger W. Dijkstra, University Texas at Austin, File:Edsger Wybe Dijkstra.jpg, CC BY-SA 3.0

2. (a), (b) Ermittle mit dem Dijkstra-Algorithmus den kürzesten Weg von A zu allen anderen Knoten.  
 \*\*(c) Ermittle mit dem Dijkstra-Algorithmus den kürzesten Weg von Ka nach Ma.

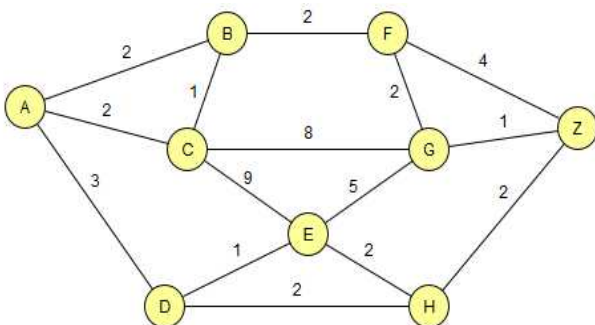
(a)



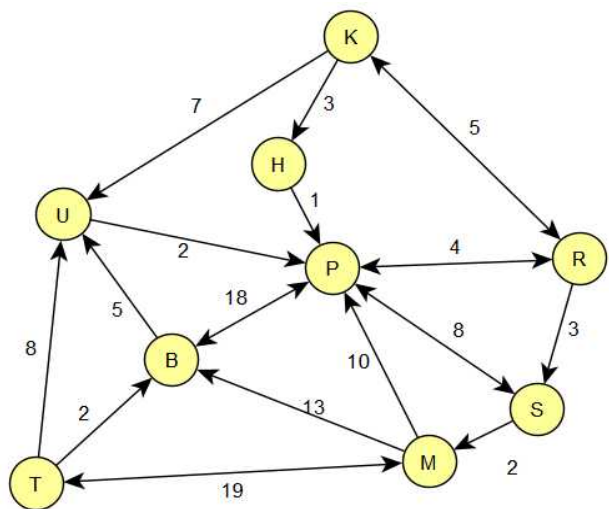
\*\* (c)



(b)



3. (a) Ein Taxifahrer verzweifelt an den vielen Einbahnstraßen in der Stadt. Für jede Straße hat er sich die Fahrzeit notiert und möchte sich nun einen Plan machen, wie er am schnellsten von seinem Taxistand zum Hotel, zum Marktplatz und zum Rathaus kommt. Hilf ihm.  
 (b) Ermittle auch die kürzesten Fahrzeiten zu Bahnhof, Uni, Kino, Sternwarte und Park.



4. In sozialen Netzwerken kann man anderen Usern folgen und andere User können einem selber folgen. Hier sind einige Jugendliche mit ihren Follower-Beziehungen:

- Anna folgt Charlotte, Fiona
- Bela folgt Charlotte, Johann
- Charlotte folgt Bela, Fiona, Daria
- Daria folgt Charlotte, Emil
- Emil folgt Daria, Fiona, Kevin
- Fiona folgt Charlotte, Emil, Hanna, Daria
- Giorgio folgt Emil
- Hanna folgt Anna, Fiona, Kevin, Giorgio
- Johann folgt Bela
- Kevin folgt Charlotte, Emil

Über wie viele Zwischenschritte folgt Anna den anderen Jugendlichen?

5. \*\* Gehe zur Seite:

[https://www-m9.ma.tum.de/graph-algorithms/spp-dijkstra/index\\_de.html](https://www-m9.ma.tum.de/graph-algorithms/spp-dijkstra/index_de.html)<sup>4</sup> und bearbeite (a) 'Forschungsaufgabe 3' und (b) 'Weiteres'.

6. \*\* Der A\*-Algorithmus (sprich: „A Stern Algorithmus“)

Gehe auf die Seite: <https://qiao.github.io/PathFinding.js/visual/><sup>5</sup>

- (a) Probiere dort den Dijkstra- und den A\*-Algorithmus aus.
- (b) Setze ein Hindernis zwischen Start und Zielknoten. Wie verhalten sich Dijkstra- und A\*-Algorithmus nun? Wie beurteilst du das Verhalten?

7. \*\* Der A\*-Algorithmus (sprich: „A Stern Algorithmus“)

Öffne die Seite: [https://www-m9.ma.tum.de/graph-algorithms/spp-a-star/index\\_de.html](https://www-m9.ma.tum.de/graph-algorithms/spp-a-star/index_de.html)<sup>6</sup>

- (a) Klicke auf den Reiter „Beschreibung des Algorithmus“ und lies die Beschreibung des A\* Algorithmus.
- (b) Klicke auf den Reiter „Forschungsaufgabe 2“ und führe den Algorithmus Schritt für Schritt aus. Vielleicht hilft es dir, denselben Graphen mit dem Dijkstra-Algorithmus zu bearbeiten. Dann gehe zu „Forschungsaufgaben 1“.
- (c) Erkläre, welche Vorteile der A\*-Algorithmus gegenüber dem Dijkstra-Algorithmus bietet.

4 Abgerufen am 28.04.19

5 Abgerufen am 28.04.19

6 Abgerufen am 28.04.19