

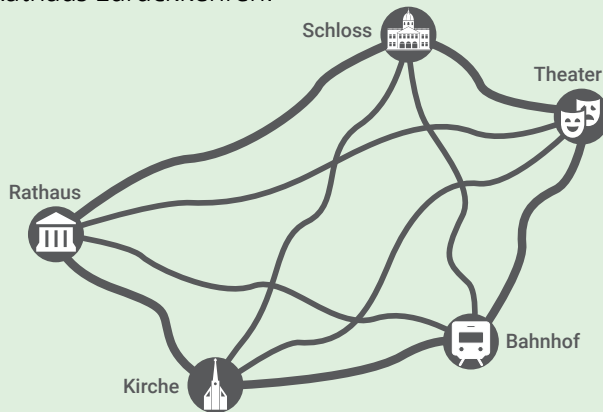
Themen

	Seite	
Grundbegriffe zu Graphen	A-2	1
Eulersche Kantenzüge	A-4	2
Königsberger Brückenproblem	A-6	3
Briefträgerproblem	A-9	4
Minimaler Spannbaum	A-12	5
Kürzeste Wegstrecke	A-14	6
Algorithmus von Dijkstra	A-16	7

Briefträgerproblem

Aufgabe 1

Ein Briefträger soll in den Straßen der abgebildeten Stadt Briefe austragen. Er soll am Rathaus starten, alle Straßen ablaufen und wieder zum Rathaus zurückkehren.



- a) Existiert in der abgebildeten Stadt ein Weg, auf dem der Briefträger alle Straßen genau einmal abläuft und der ihn wieder zum Rathaus zurück führt? Begründe deine Antwort.

Da alle Knoten einen geraden Grad haben, enthält der Graph einen Eulerkreis. Eulerkreise sind geschlossene Kantenzüge, die alle Kanten eines Graphen genau einmal enthalten und bei denen Start- und Endknoten identisch sind. Folglich existiert ein Weg, auf dem der Briefträger alle Straßen genau einmal abläuft und wieder zurück zum Rathaus gelangt.

- b) Wie ermittelt man die Länge des kürzesten Weges, den der Briefträger durch die Stadt nehmen kann?

Eulerkreise enthalten alle Kanten eines Graphen genau einmal. Die Länge des kürzesten Weges in einem Eulerkreis ermittelt man folglich durch Addieren der Länge aller Kanten (Straßen).

- c) Nenne einen möglichen Weg, den der Briefträger durch die Stadt nehmen kann.

Rathaus – Theater – Kirche – Schloss – Bahnhof – Rathaus – Schloss – Theater – Bahnhof – Kirche – Rathaus

Aufgabe 2

Nenne weitere Anwendungsbeispiele für das Briefträgerproblem?

- Tourenplan Schneepflug
- Routenplan Google-Street-View-Auto
- Tourenplanung Müllabfuhr

Aufgabe 3

Warum ist es sinnvoll, in einem Graphen durch das Einfügen zusätzlicher Kanten einen Eulerkreis zu erzeugen?

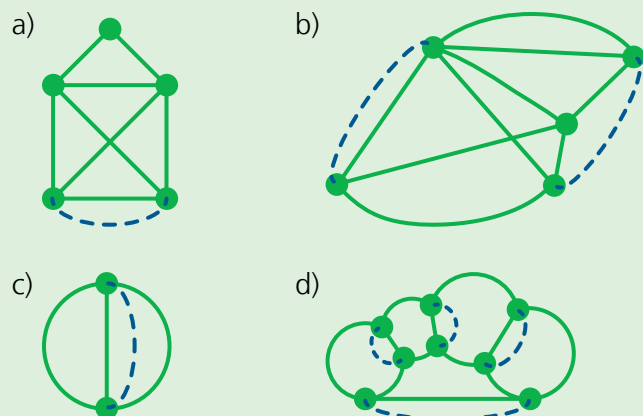
Ein Graph ist ein Eulerkreis, wenn in ihm ein Weg möglich ist, der alle Kanten genau einmal durchläuft und zum Ausgangspunkt zurück führt.

Durch das Erzeugen eines Eulerkreises mit Hilfe zusätzlicher Kanten ist ein Weg möglich, auf dem der Briefträger alle Straßen (Kanten) einschließlich der zusätzlichen Kanten nur einmal ablaufen muss und wieder zurück zum Ausgangspunkt gelangt.

Aufgabe 4

Durch welche zusätzlichen Kanten lassen sich in den folgenden Graphen Eulerkreise erzeugen? Zeichne die zusätzlichen Kanten ein.

Beispiellösungen:

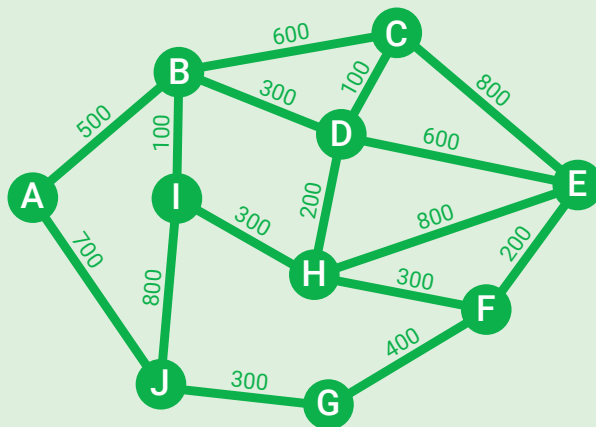


Briefträgerproblem

Aufgabe 5

Der Graph bildet eine Stadt ab, in deren Straßen ein Briefträger die Post austragen soll, bevor er wieder zum Ausgangspunkt zurückkehrt.

Der Graph soll verändert werden, um die kürzeste Route des Briefträgers ermitteln zu können.



- a) Welche Knoten müssen dazu mit zusätzlichen Kanten verbunden werden?

Die Knoten C, F, I, J sind 3. Grades. Um einen Eulerkreis zu erhalten, müssen diese vier Knoten mit zusätzlichen Kanten verbunden werden.

- b) Welche Möglichkeiten gibt es, diese Knoten mit zusätzlichen Kanten zu verbinden?

$$I-J \quad + \quad C-D-H-F$$

$$I-J \quad + \quad C-E-F$$

$$F-G-J + C-B-I$$

$$F-G-J + C-D-B-I$$

$$F-G-J + C-D-H-I$$

- c) Welche Verbindungen ergeben zusammen die kürzeste Wegstrecke?

$$I-J \quad + \quad C-D-H-F = 800 + 600$$

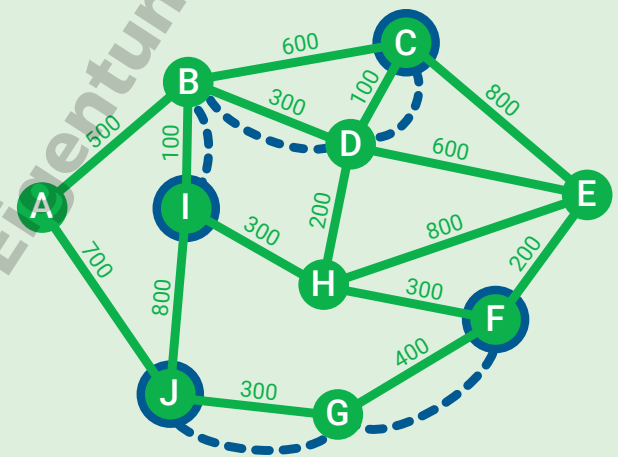
$$I-J \quad + \quad C-E-F = 800 + 1000$$

$$F-G-J + C-B-I = 700 + 700$$

$$F-G-J + C-D-B-I = 700 + 500$$

$$F-G-J + C-D-H-I = 700 + 600$$

Die kürzeste Wegstrecke ergibt sich durch die Verbindungen $F-G-J$ und $C-D-B-I$ mit einem Gesamt-Kantengewicht von 1200.



Eulerkreis finden mit dem Hierholzer-Algorithmus

Auf der Website der Technischen Universität München gibt es die Möglichkeit, mit Hilfe des Algorithmus von Hierholzer in einem selbst erstellten Graphen einen Eulerkreis zu finden.

Autoren: Mark J. Becker, Wolfgang F. Riedl

https://www-m9.ma.tum.de/graph-algorithms/hierholzer/index_de.html (Stand März 2021)

Algorithmus von Dijkstra

Der niederländische Informatiker Edsger W. Dijkstra (1930–2002) erfand einen Algorithmus, der das Problem des kürzesten Pfades löst. Mit Hilfe des Algorithmus kann man – ausgehend von einem Startknoten – den kürzesten Pfad zu allen anderen Knoten eines kantengewichteten Graphen bestimmen.

Der Algorithmus basiert darauf, dass man vom Startknoten und von jedem weiteren besuchten Knoten aus stets derjenigen Kante folgt, mit der sich die kürzeste Verbindung zwischen dem Startknoten und dem nächsten erreichbaren, unbesuchten Knoten aufbauen lässt.

Entsprechend wählt man in unserem Beispiel unten im 3. Schnitt die Kante mit dem Kantengewicht 2, und nicht die Kante mit dem Kantengewicht 5.

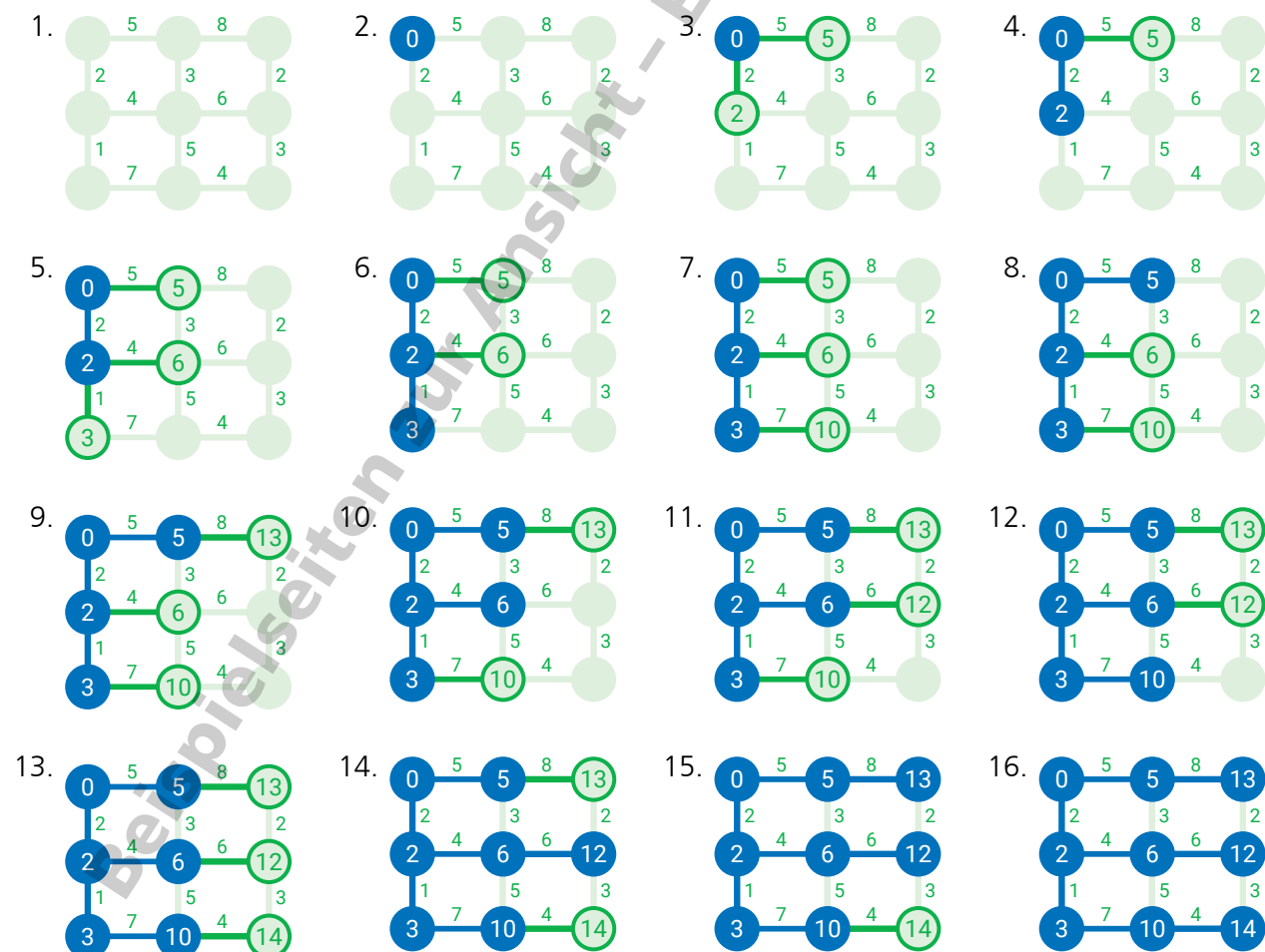
Der Algorithmus von Dijkstra zählt zu den so genannten gierigen Algorithmen. Bei diesen Algorithmen wird in jedem Schritt nur die aktuell beste Lösung ausgewählt. Vorherige oder nachfolgende Entscheidungen werden nicht berücksichtigt.

Dementsprechend wird die Entfernung zu einem besuchten Knoten (in unserem Beispiel blau gefärbt) nicht mehr geändert, auch wenn sich in der Folge eine geringere Entfernung ergibt.

Die Entfernungen zu Knoten, die noch nicht besucht wurden (erkennbar am blauen Rand), können sich hingegen ändern, sobald sich über eine andere Route ein kürzerer Weg dorthin ergibt.

So wird schrittweise Knoten für Knoten besucht, bis der kürzeste Weg zum Zielknoten gefunden wurde oder bis die Entfernungen aller Knoten zum Startknoten bekannt sind.

Beispiel für das Anwenden des Algorithmus von Dijkstra



Algorithmus von Dijkstra

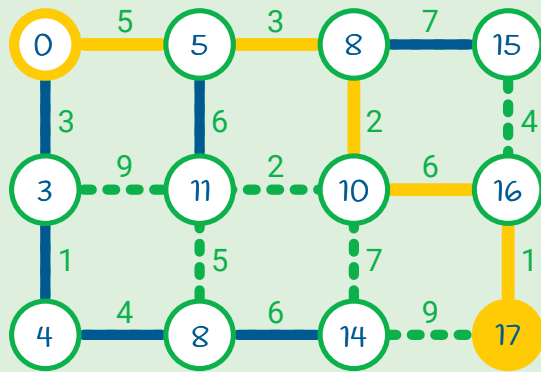
Aufgabe 1

Was zeichnet die so genannten gierigen Algorithmen aus?

Bei gierigen Algorithmen wird in jedem Schritt nur die aktuell beste Lösung ausgewählt. Vorherige oder nachfolgende Entscheidungen werden nicht berücksichtigt.

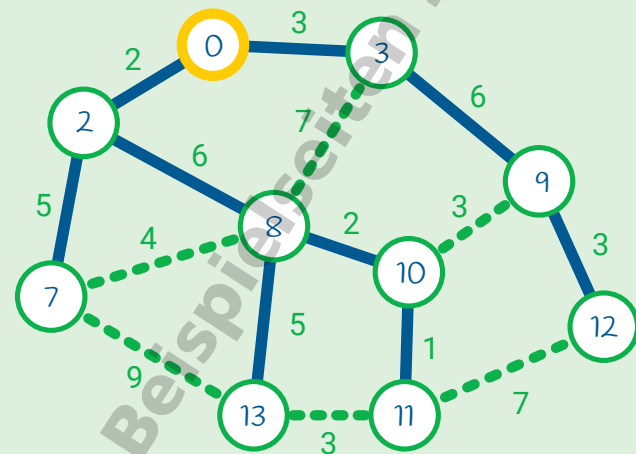
Aufgabe 2

Ermittle mit Hilfe des Algorithmus von Dijkstra den kürzesten Pfad vom gelb umrandeten Knoten zum gelben Knoten.



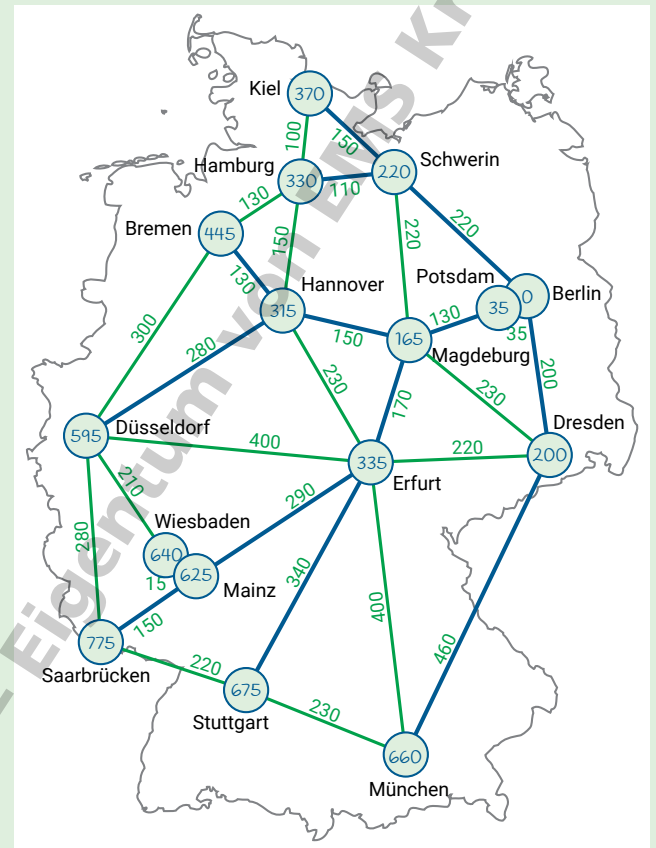
Aufgabe 3

Ermittle mit Hilfe des Algorithmus von Dijkstra die Länge der kürzesten Pfade vom gelb umrandeten Knoten zu allen übrigen Knoten des Graphen.



Aufgabe 4

Ermittle mit Hilfe des Algorithmus von Dijkstra die Länge der kürzesten Routen von der Bundeshauptstadt Berlin zu allen Landeshauptstädten.



Aufgabe 5

Worin besteht der Vorteil des Algorithmus von Dijkstra gegenüber der Brute-Force-Methode?

Mit der Brute-Force-Methode benötigt man selbst bei kleinen Graphen sehr viel Zeit, um alle möglichen Wege durch den Graphen zu finden und den kürzesten Weg zu ermitteln.

Beim Algorithmus von Dijkstra reichen im Vergleich dazu wenige Schritte aus, um den kürzesten Weg durch einen Graphen zu finden.