

## DOJOAUFGABE: GRAPHEN ALS DATENSTRUKTUR

# Kürzeste Wege finden

Ein zentrales Thema der Informatik sind Algorithmen und Datenstrukturen. Diesmal geht es um Graphen als Datenstruktur und einen sehr wichtigen darauf basierenden Algorithmus.

Als regelmäßiger Leser der dotnetpro erinnern Sie sich vielleicht daran, dass Graphen vor vielen Jahren bereits Thema einer Übungsaufgabe waren. Wiederholung schadet nicht beim Lernen, allemal nach so vielen Jahren. Daher geht es im vor Ihnen liegenden Monat erneut um das Thema Graphen als Datenstruktur.

Ein Graph besteht aus Knoten und Kanten. Eine Kante verbindet zwei Knoten. Bei gerichteten Kanten ist ein Knoten der Start-, der andere der Zielknoten. Bei ungerichteten Kanten gibt es diese Unterscheidung nicht.

Eine praktische Anwendung für einen Graphen ist beispielsweise das Schienennetz der Deutschen Bahn. Jeder Bahnhof wird in einem solchen Graphen als Knoten dargestellt, Verbindungen zwischen den Bahnhöfen als Kanten. Die erste Herausforderung besteht nun darin, eine Datenstruktur zu implementieren, mit der man Knoten und Kanten speichern kann. Die zweite Herausforderung besteht darin, Algorithmen zu finden, die auf Graphen arbeiten, und diese dann zu implementieren.

Für das Beispiel eines Schienennetzes fehlt noch ein weiteres Konzept, nämlich das der gewichteten Kanten. Dabei hat jede Kante ein sogenanntes Gewicht, welches in Algorithmen mit einbezogen werden kann. Bei einem Schienennetz kann beispielsweise die Entfernung zwischen zwei Bahnhöfen als Gewicht der verbindenden Kante gespeichert werden. So kann ein Suchalgorithmus nicht irgendeinen Weg zwischen zwei Bahnhöfen suchen, sondern den kürzesten. Dabei bezieht sich der „kürzeste Weg“ auf die Kantengewichtung. Ergebnis einer Suche ist also ein Pfad zwischen zwei Knoten, bei dem die Summe der Kantengewichte minimal ist. Als Kantengewicht kann anstelle der Entfernung natürlich auch die Fahrzeit herangezogen werden.

Zum Problem des kürzesten Pfades in einem gewichteten Graphen hat Edsger Dijkstra bereits 1959 eine Lösung vorgestellt. Der nach ihm benannte Dijkstra-Algorithmus [1] sucht ausgehend von einem Startknoten die Pfade zu allen anderen Knoten. Ergebnis ist ein Baum von kürzesten Pfaden zu allen anderen Knoten.

## Aufgabe

Die Übungsaufgabe besteht somit darin, sich zunächst etwas in den Dijkstra-Algorithmus einzulesen. Ein Grundverständnis für Graphen und den Algorithmus sollte vorhanden sein, bevor damit begonnen wird, Graphen als Datenstruktur zu implementieren. Da Graphenalgorithmen häufig von der konkreten Repräsentation des Graphen abhängen, ist es

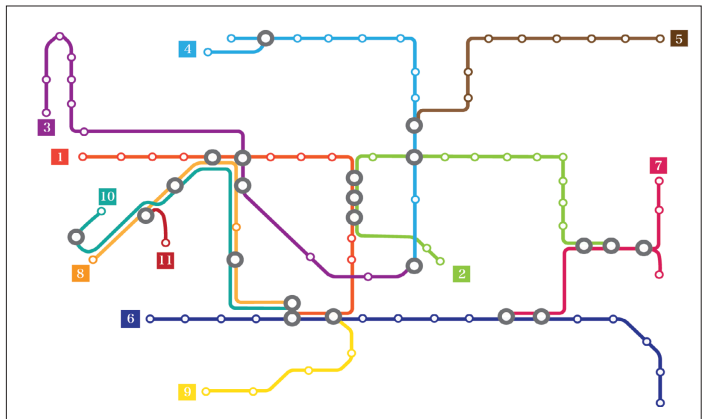


Foto: Shutterstock / Leah Lee

sinnvoll, sich vor der Implementation der Datenstruktur bereits mit dem Algorithmus zu befassen. Des Weiteren ist es erforderlich, sich ein geeignetes API zu überlegen.

Am besten gehen Sie die Aufgabe zunächst so konkret wie möglich an: Versuchen Sie, ein Schienennetz als Graph darzustellen. Als Kantengewicht wird die Fahrzeit in Minuten verwendet. Anschließend überlegen Sie sich ein geeignetes API, mit dem das Schienennetz definiert wird. Daran anschließend muss dann der Dijkstra-Algorithmus implementiert werden, um Verbindungen zwischen zwei Bahnhöfen zu suchen. Ergebnis ist eine Liste möglicher Verbindungen mit der Gesamtfahrzeit. Die Zwischenhalte sollten natürlich ebenfalls Bestandteil des Ergebnisses sein.

Wenn eine Implementation für eine solche Verbindungssuche vorliegt, können Sie überlegen, wie das API verallgemeinert werden kann, um es auch in einem anderen Kontext verwenden zu können.

Ich wünsche viel Freude mit den Graphen. ■

[1] <https://de.wikipedia.org/wiki/Dijkstra-Algorithmus>



### Stefan Lieser

sucht ständig nach Verbesserung und neuen Wegen, um die innere Qualität von Software zu optimieren. Gemeinsam mit Ralf Westphal hat er die Clean Code Developer Initiative (<https://clean-code-developer.de>) ins Leben gerufen.

<https://lieser-online.de>

dnpcode

A2011dojoAufgabe