

BIG B4NG Challenge, 20. Wettbewerb Aufgabe 2

Diese Aufgabe wird vom Institut für Kartographie und Geoinformatik der Leibniz Universität Hannover gestellt. Weitere Informationen zum Studiengang Geodäsie und Geoinformatik findet ihr unter gug.uni-hannover.de

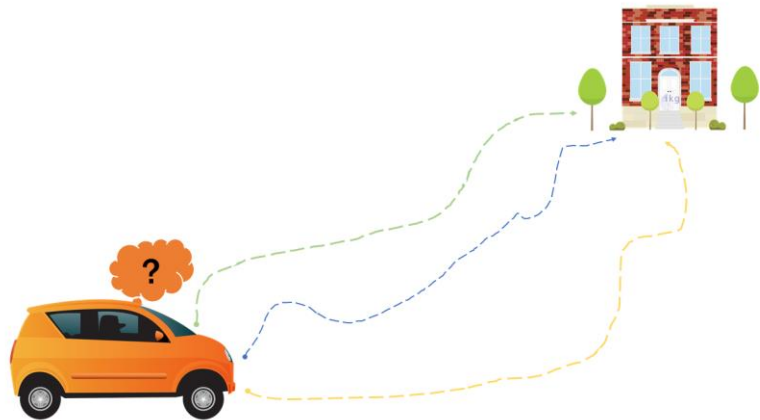
Navigation

In dieser Aufgabe geht es um das Finden der eigenen Position und die Wegfindung.

Im ersten Aufgabenteil geht es um die allgemeine Positionsbestimmung mittels global verfügbarer Navigationssatellitensysteme. Darüber hinaus soll untersucht werden, was die Einflüsse auf die Genauigkeit der Positionsbestimmung sind.

Eine erste Einführung in das Thema der Positionierung findet ihr im folgenden Video unseres Kollegen vom Institut für Erdmessung:

<https://www.youtube.com/watch?v=m0aqWUy88yc>



a) Positionierung (8 Punkte)

1. Welche globalen Navigationssatellitensysteme (GNSS) gibt es? Wie viele Satelliten haben diese jeweils und wie viele sind derzeit in der aktiven Nutzung?
2. Beschreibt kurz in eigenen Worten, wie das Grundprinzip der GNSS-Positionierung funktioniert, und fertigt eine Skizze (ohne Störeinflüsse) dazu an. Beachtet dabei folgende Punkte: Wie viele Satelliten werden mindestens benötigt? Zeichnet die Pfade der Signale ein.
3. Nennt 3 Störeinflüsse, die das Ergebnis der Positionierung beeinflussen. Hinweis: An welchen Stellen und wovon kann das Signal auf dem Weg vom Satelliten zum Empfänger absorbiert, gestört oder abgelenkt werden?
4. Mit welcher Genauigkeit (in cm) kann die Position mit einer "Profi-Ausrüstung" bestimmt werden? Und wie schneidet dagegen das eigene Smartphone ab? Hinweise hierzu können im Video zu Positionierung gefunden werden.

Im zweiten Aufgabenteil geht es um die Navigation bzw. um die Wegfindung. Bei der Wegfindung handelt es sich um das sogenannte Routing, was einen wichtigen Teil der Navigation ausmacht. In der Navigation geht es insbesondere um die Findung einer optimalen Route zu einem bestimmten Ziel, aber auch um das Befolgen dieser Route. Eine optimale Route kann von verschiedenen Faktoren abhängen, die durch den Nutzer bestimmt werden können.

In den folgenden Aufgabenteilen b) und c) geht es um das Routing von Fahrzeugen mit Hilfe von verschiedenen Routing-Algorithmen. In diesem Fall soll beim Routing die kürzeste Strecke von der Stadt Stolzenau zu der Stadt Salzgitter gefunden werden. Diese sogenannten Routing-Algorithmen agieren auf Straßennetzen, sogenannten gewichteten Graphen. Diese bestehen aus Knotenpunkten (hier: Städte), welche über gewichtete Kanten (hier: Straßen) miteinander verbunden sind. In dieser Aufgabe betrachten wir die Gewichte der Kanten als (Straßen-)Distanzen, die zur Verbindung zweier Knoten zurückgelegt werden müssen. Aber natürlich könnten auch andere Kosten als Gewichte für die Kanten genutzt werden. Ein Beispiel eines solchen Graphen findet ihr im Aufgabenteil c).

b) Routing-Algorithmen – Theorie (11 Punkte)

1. Welche Einflussfaktoren (Kantengewichte) kann es bei der Berechnung einer Route geben? Ergänzt die Tabelle und nennt 4 Einflussfaktoren.

Einflussfaktor	Beschreibung
Straßendistanz	Abstand in Straßenkilometern

2. Nennt zusätzlich 3 Beispiele für Routing-Algorithmen, um den Weg von Stolzenau nach Salzgitter zu finden. Gebt für jeden Algorithmus einen Vor- bzw. Nachteil an. Vervollständigt die Tabelle.

Routing-Algorithmen	Vorteile	Nachteile
Dijkstra		

In den nachfolgenden Teilaufgaben wird der Dijkstra-Algorithmus genauer betrachtet.

3. Beschreibt die Arbeitsschritte des Dijkstra-Algorithmus in eigenen Worten. Hinweis: Vergeßt das Abbruchkriterium nicht.

4. Warum darf man beim Dijkstra-Algorithmus aufhören, weitere Knoten zu betrachten, sobald man das Ziel gefunden hat?
5. Oskar wohnt in Hannover und hat kein Ziel an diesem Tag. Er hat aber noch Sprit für 250 km und will wissen, welche Orte er damit besuchen könnte. Erläutert kurz, wie er das herausfinden kann.
6. Für die kürzeste Route ist es egal, ob man den Dijkstra-Algorithmus vom Start zum Ziel oder vom Ziel zum Start laufen lässt. Fahrer befolgen manchmal die vorgeschlagene Route nicht. Sind sie falsch abgebogen, brauchen sie schnell eine neue Route. Was ist in diesem Fall besser: wenn das Dijkstra-Navigationssystem die originale Route vom Start zum Ziel oder rückwärts vom Ziel zum Start berechnet hat? Gebt eine kurze Begründung für die Entscheidung an.

Nachdem der Dijkstra-Algorithmus theoretisch behandelt wurde, soll dieser im letzten Aufgabenteil praktisch angewandt werden. Das nachfolgende Video gibt eine Einführung in die Anwendung des Dijkstra-Algorithmus: <https://www.youtube.com/watch?v=JtJfcJD4QmY>

Im nachfolgendem Beispiel soll die optimale (kürzeste) Route von Stolzenau nach Salzgitter ermittelt werden. Dazu sollen zu Beginn erst der Dijkstra-Algorithmus und anschließend der Prim-Algorithmus angewandt werden. Die erzielten Ergebnisse beider Routing-Algorithmen sollen anschließend verglichen werden. Hinweis: Für die Anwendung des Prim-Algorithmus (Frage 4) wird zusätzlich zum Graphen in Abbildung 1 auch Tabelle 1 mit den Luftliniendistanzen von allen Knoten nach Salzgitter benötigt. Beachtet darüber hinaus, dass die angegebenen Distanzen keine Abbildung der realen Verhältnisse darstellen.

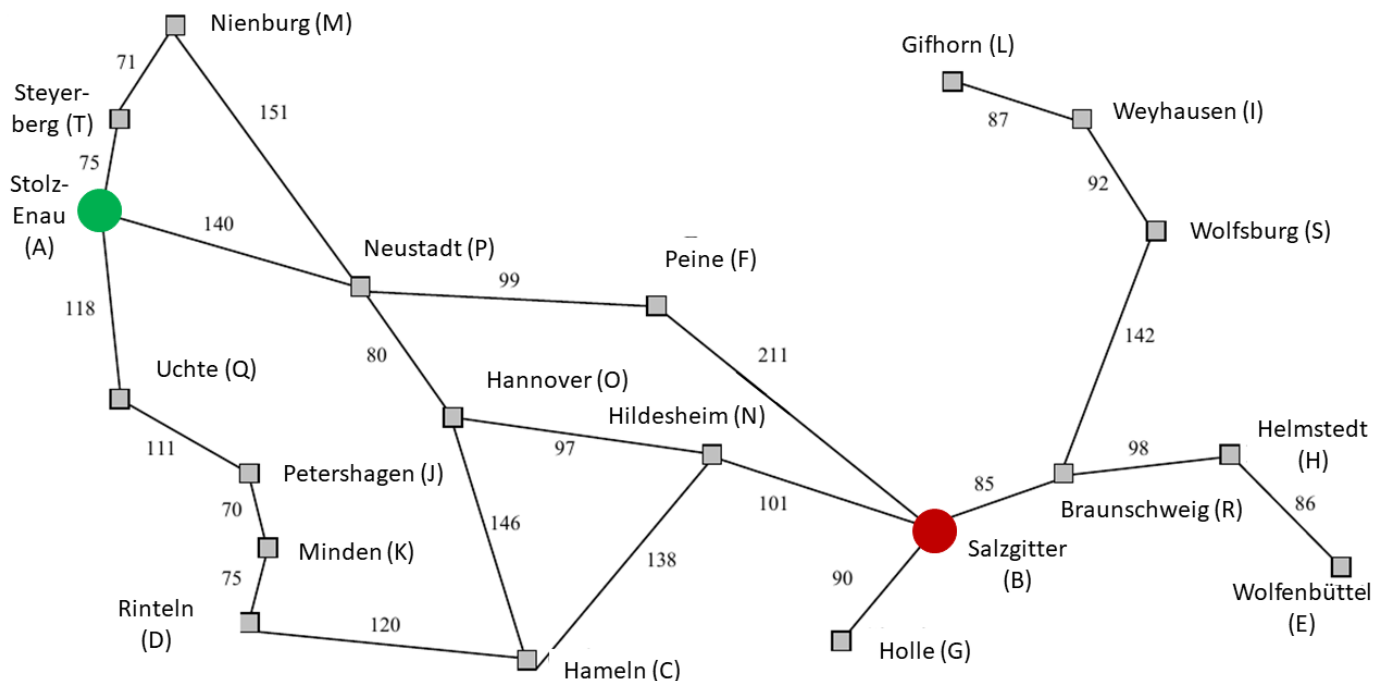


Abbildung 1: Städtegraph mit erfundenen Straßendistanzen (in km).

A	Stolzenau	366	B	Salzgitter	0	C	Hameln	160	D	Rinteln	242
E	Wolfenbüttel	161	F	Peine	178	G	Holle	77	H	Helmstedt	151
I	Weyhausen	226	J	Petershagen	244	K	Minden	241	L	Gifhorn	234
M	Nienburg	380	N	Hildesheim	98	O	Hannover	193	P	Neustadt	253
Q	Uchte	329	R	Braunschweig	80	S	Wolfsburg	199	T	Steyerberg	374

Tabelle 1: Erfundene Luftliniendistanzen (in km) zur Stadt Salzgitter.

c) Routing Algorithmen – Praxis (11 Punkte)

1. Führt den Dijkstra-Algorithmus auf dem dargestellten Graphen (Abbildung 1) aus, um den kürzesten Weg von Stolzenau nach Salzgitter zu finden. Listet alle Zwischenschritte in der beigefügten Tabelle auf (BigBangChallenge_2020_A2_Geodasie_V2_Tabelle_C1.xlsx). Wie lang ist die ermittelte Strecke?
2. Zeichnet die kürzeste Strecke (blau) in den Graphen ein und markiert alle vom Algorithmus betrachteten Städte (gelb).
3. Beschreibt kurz das allgemeine Vorgehen der Greedy-Algorithmen (z. B. Prim) in eigenen Worten.
4. Führt den Prim-Algorithmus auf dem dargestellten Graphen aus, um den kürzesten Weg von Stolzenau nach Salzgitter zu finden. Nutzt hierbei die Luftliniendistanzen als Kantengewichte. Listet alle Zwischenschritte tabellarisch auf, verwendet unten stehende Tabelle, wo der erste Schritt schon durchgeführt wurde. Wie lang (basierend auf den Kantengewichten in Abbildung 1) ist die ermittelte Strecke?

BETRACHTET	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
A																253	329			374
A, P																				

5. Zeichnet die Lösung des Prim-Algorithmus in den Graphen ein. Markiert die ermittelte Strecke (rot) und umkreist alle vom Algorithmus betrachteten Städte (grün).
6. Vergleicht die Ergebnisse des Dijkstra-Algorithmus und des Prim-Algorithmus. Welcher Algorithmus gibt das bessere Ergebnis zurück? Fasst die Erkenntnisse kurz zusammen.

Viel Erfolg bei dieser zweiten Aufgabe!

Allgemeine Hinweise

Einsendeschluss: Sonntag, 22. November 2020, 19:59 Uhr.

Gebt eure Lösungen über unser Portal ab: <https://portal.studienberatung.uni-hannover.de/anmeldungen/users/login>

Zulässige Dateiformate sind: PDF für die zusammengeschriebene Lösung (mit eingebetteten Bildern) sowie unter Windows gängige Videoformate, die sich ohne Installation von zusätzlicher Software abspielen lassen, z. B. mp4.

Die Dateien sollten nicht größer als 7,5 MB sein (die Dateien können gezippt sein)! Bitte gebt auch euren Teamnamen, die Namen der Gruppenmitglieder sowie deren Schulen an. Bitte benennt eure hochgeladenen Dateien nach dem Gruppennamen.

ACHTUNG bei Zip-Dateien! Um sicherzugehen, dass eure Dateien wirklich fehlerfrei und für die Korrektor*innen zu öffnen sind, solltet ihr eure Zip-Dateien etc. noch mal von eurem Account herunterladen und öffnen. Dateien, die sich nicht öffnen lassen, können nicht bewertet werden!

Die Teilnahmebedingungen und weitere Informationen findet ihr unter:

www.uni-hannover.de/bigbangchallenge

Der Rechtsweg ist ausgeschlossen.