4 Science Challenge, 25. Wettbewerb

Aufgabe 2: Mathematik

03. November 2025

Diese Aufgabe kommt aus dem Bereich der Mathematik. Weitere Informationen zum Studiengang Mathematik findet ihr unter `https://www.maphy.uni-hannover.de/de/studium/im-studium/mathematik`.

Einführung

Oft wollen wir Menschen verschiedene Objekte und ihre Verbindungen untereinander verstehen. Ein Beispiel für ein solches Netzwerk ist das Schienennetz der DB. Dabei stellt man sich als Fahrgast natürlich Fragen wie "Was ist der schnellste Weg?", "Was ist der günstigste Weg?", aber auch als Manager der Bahn "Wie saniere ich Bahnstrecken möglichst kosteneffizient?". Diese Art von Fragen wollen wir nun mathematisch anhand der Graphentheorie untersuchen.

Kapitel 1: Grundlagen – Was ist ein Graph? (6 Punkte)

Einführung

Wichtige Begriffe

Graph: Mathematische Struktur G(V, E) einer Menge von Objekten (Knoten V: Bahnhöfe) und ihrer Verbindungen (Kanten E: Bahnverbindungen)
Grad eines Knotens: Anzahl der existierenden Verbindungen zu einem Knoten
Gewichteter Graph: Jede Kante hat ein Gewicht (z. B. Entfernung in km, Fahrtzeit in min, Preis in €)
Ungerichteter Graph: Strecken können in beide Richtungen befahren werden
Multigraph: In einem Multigraph sind mehrere Kanten zwischen 2 Knoten erlaubt. Ein Graph, bei dem nur eine Kante zwischen 2 Knoten existiert, wird **einfacher Graph** genannt. Ebenfalls sind nur hier Schleifen, also Kanten von einem Knoten auf sich selbst, erlaubt.

Aufgaben zum Graphenzeichnen

Gegeben sind folgende Bahnlinien:

- **ICE1:** Hamburg ↔ Hannover ↔ Göttingen ↔ Frankfurt ↔ Stuttgart
- **ICE2:** Aachen ↔ Köln ↔ Hannover ↔ Berlin
- **ICE3:** München ↔ Erfurt ↔ Dresden ↔ Berlin
- **Regio1:** Hamburg ↔ Bremen ↔ Osnabrück ↔ Münster ↔ Köln ↔ Aachen
- **Regio2:** Bremen ↔ Hannover ↔ Magdeburg ↔ Erfurt ↔ Göttingen

Aufgabe 1.1: Eigenschaften des Graphen (4 Punkte)

a) Zeichnet diesen Graphen auf Papier. Orientiert die Knoten grob an ihrer geographischen Lage. (**2 Punkt**)

b) Wie viele Knoten und Kanten hat der Graph? Handelt es sich um einen Multi- oder einfachen Graphen? (**1 Punkt**)

c) Bestimmt den Grad jedes Knotens. Welcher Bahnhof ist der wichtigste Knotenpunkt? (**1 Punkt**)

Aufgabe 1.2: Distanzen (2 Punkte)

Ergänzt in eurem Graphen die Entfernungen zwischen den Städten (Luftlinie). Nutzt dafür `https://www.luftlinie.org/` und rundet auf ganze Kilometer.

Beispiel

Beispiele: Hamburg–Bremen: 95 km, Bremen–Osnabrück: 103 km, Hannover–Göttingen: 94 km

Kapitel 2: Wege und Pfade (11 Punkte)

Einführung

Wichtige Begriffe

Weg: Folge von Knoten, die durch Kanten verbunden sind (Knoten dürfen wiederholt werden)
Pfad: Weg, bei dem jeder Knoten höchstens einmal besucht wird
Länge eines Weges: Anzahl der Kanten
Gewicht eines Weges: Summe aller Kantengewichte (z. B. Gesamtdistanz)
Zusammenhängender Graph: Zwischen je zwei Knoten existiert mindestens ein Weg
Zyklus: Weg, der am gleichen Knoten beginnt und endet (ohne Knotenwiederholungen)
Robustheit: Die Fähigkeit eines Graphen, zusammenhängend zu bleiben, wenn man Kanten oder Knoten entfernt.

Aufgaben zur Pfadfindung

Aufgabe 2.1: Pfade erkunden (4 Punkte)

Verwendet euren Graphen aus Kapitel 1.

a) Findet alle möglichen Pfade von Aachen nach Hamburg. Listet diese auf. (**2 Punkte**)

b) Welcher Pfad hat die wenigsten Zwischenstopps? (**1 Punkt**)

c) Welcher Pfad hat die kürzeste Gesamtentfernung? Berechnet diese. (**1 Punkt**)

Aufgabe 2.2: Netzstörungen (3 Punkte)

a) Ist der Graph zusammenhängend? Begründet aus Sicht einer Person, die ein solches Bahnnetz konzipiert. (**1 Punkt**)

b) Hannover wird wegen eines Unfalls komplett gesperrt. Ist der Graph noch zusammenhängend? Trefft eine Aussage über die Robustheit des Graphen. Durch welche einfache Verbindung kann man diese verbessern? (**1 Punkt**)

c) Gibt es Bahnhöfe, die ebenfalls kritisch sind? (**1 Punkt**)

Aufgabe 2.3: Rundfahrten (4 Punkte)

a) Findet Zyklen in dem Graphen. Wie viele verschiedene Zyklen könnt ihr finden? (**3 Punkte**)

b) Ein Touristenzug startet in Bremen und soll möglichst viele verschiedene Städte besuchen, bevor er nach Bremen zurückkehrt. Wie lang ist diese Route und wie viele Städte besucht sie? (**1 Punkt**)

Kapitel 3: Optimierungsprobleme (8 Punkte)

Einführung - Optimale Pfade

Wir haben in Aufgabe 2 bereits die möglichen Pfade von Aachen nach Hamburg untersucht und dabei nach der Anzahl der Zwischenstopps und der Distanz untersucht. Diese beiden Größen sind aber natürlich nicht die einzigen, die für einen Fahrgast wichtig sind. So spielen Preis und Fahrtzeit ebenfalls eine Rolle, die je nachdem unterschiedlich sein können. Wir nehmen daher an, dass ICE-Züge mit **150 km/h** fahren und ein Fahrtticket ca. **0,15 €/km** kostet. Bei einer Regioverbindung ist die Geschwindigkeit **90 km/h**, dafür sind sie allerdings auch billiger mit **0,07 €/km**. Dementsprechend verändern sich die Kantengewichte, je nachdem, welche Größe wir betrachten. Wir wollen nun den optimalen Weg ähnlich zu Aufgabe 2.1 **von Aachen nach Hamburg** finden. Diesmal wollen wir das aber nicht per Hand machen, sondern einen Algorithmus betrachten, welcher uns dabei unterstützen kann. Denn für komplizierte Graphen kann es sehr schnell sehr komplex werden, die optimalen Pfade per Hand zu bestimmen. Dafür verwenden wir den sogenannten Dijkstra-Algorithmus. Dieser hilft dabei, den kürzesten, schnellsten oder billigsten Pfad zu finden. Normalerweise werden diese Algorithmen in Programmen verwendet, die deutlich schneller rechnen können als wir das per Hand können. Wir wollen uns hier aber trotzdem einmal die Funktionsweise anschauen.

Dijkstra-Algorithmus

Ziel: Findet den kürzesten/schnellsten/günstigsten Weg von A nach B.
Idee:

i. Der Startknoten A bekommt das Gewicht 0.

ii. Alle anderen Knoten bekommen das Gewicht ∞.

iii. Besucht den Knoten mit dem kleinsten bekannten Gewicht.

iv. Aktualisiert Gewichte der benachbarten Knoten X anhand der dazugehörigen Kanten.
   - Summiert die Kantengewichte von A bis X.
   - Alle nicht benachbarten Knoten behalten das Gewicht ∞.

v. Besucht den Knoten mit dem kleinsten bisher bekannten Gesamtgewicht.
   - Auch vorherige Knoten berücksichtigen!

vi. Wiederholt die Schritte iv. und v., bis B erreicht ist.

Tipp: Im Anhang ist ein ausführliches Beispiel zu dem Verfahren.
Wichtig: Funktioniert nur bei nicht negativen Graphen!

Aufgaben zu optimalen Wegen

Aufgabe 3.1: Schnellste vs. kürzeste Route (4 Punkte)

Vereinfachtes Bahnnetz

In diesem Aufgabenteil müssen die Knoten (und ihre Kanten) **Erfurt, Magdeburg, Frankfurt, Stuttgart, München** und **Dresden NICHT** betrachtet werden.

a) Wendet den Dijkstra-Algorithmus an, um die billigste und die schnellste Route von Aachen nach Hamburg zu finden. Es genügt die Angabe der jeweiligen Tabelle (vgl. Anhang). Sind der billigste und der schnellste Pfad der gleiche?

Einführung - Minimale Netze

In Deutschland sind viele Bahnstrecken sanierungsbedürftig, jedoch kostet dies viel Geld. Zur Kostenoptimierung will die Bahn daher nur die minimale Anzahl an Strecken bzw. Streckenlänge sanieren, die benötigt werden, um alle Städte miteinander zu verbinden. In der Graphentheorie ist das der minimale Spannbaum des Graphen. Zur Bestimmung wird der Kruskal-Algorithmus verwendet.

Kruskal-Algorithmus

Ziel: Verbindet alle Bahnhöfe mit minimalem Gesamtgewicht, zum Beispiel der minimalen Gesamtlänge. Das nennt man den **minimalen Spannbaum**.
Idee:

i. Sortiert alle Kanten nach Gewicht (aufsteigend).

ii. Fügt nacheinander Kanten hinzu, die keinen Zyklus erzeugen.

iii. Stoppt bei $n-1$ Kanten (für n Knoten).

Eigenschaft: Ein Spannbaum mit n Knoten hat immer genau $n-1$ Kanten.

Aufgaben zu minimalen Netzen

Aufgabe 3.2: Sanierung der Bahnstrecken (4 Punkte)

a) Bestimmt den minimalen Spannbaum unseres Bahnnetzes aus Aufgabe 1 anhand der Streckenlängen. (**2 Punkte**)

b) Die Sanierungskosten betragen:
   - ICE-Strecke: 0,5 Mio €/km
   - Regio-Strecke: 0,2 Mio €/km

   Wie viel kostet die gesamte Sanierung? (**1 Punkt**)

c) Was ist das Problem eines minimalen Spannbaums aus Fahrgastsicht? (**1 Punkt**)

Kapitel 4: Euler- und Hamiltonwege (5 Punkte)

Einführung

Eulerpfade und Eulerkreise

Eulerpfad: Durchläuft jede *Kante* genau einmal (Knoten dürfen wiederholt werden). Dafür dürfen genau 2 oder kein Knoten einen ungeraden Grad besitzen.
Eulerkreis: Eulerweg, der am Startknoten endet. Dafür darf kein Knoten einen ungeraden Grad besitzen.
Wichtig: Der Graph muss zusammenhängend sein!
Anwendung: Wartungsteam kontrolliert jede Strecke genau einmal

Hamiltonpfade und Hamiltonkreise

Hamiltonpfad: Besucht jeden *Knoten* genau einmal (manche Kanten werden nicht benutzt)
Hamiltonkreis: Hamiltonpfad, der am Startknoten endet
Wichtig: Im Gegensatz zum Eulerpfad bzw -kreis gibt es kein einfaches Kriterium, um zu entscheiden, ob ein Hamiltonpfad bzw. -kreis existiert! Man muss ausprobieren.
Anwendung: Tourist besucht jeden Bahnhof genau einmal (oder bei Hamiltonkreis **Traveling Salesman Problem**)

Unterschied Euler vs. Hamilton

- **Euler:** Jede *Kante* einmal → Knoten dürfen mehrfach vorkommen
- **Hamilton:** Jeden *Knoten* einmal → Kanten dürfen fehlen

Aufgabe 4.1: Eulerpfade und Eulerkreise (3 Punkte)

a) Warum können in unserem Bahnnetz keine Eulerpfade oder Eulerkreise vorliegen? (**1 Punkt**)

b) Was müsste sich verändern, damit ein Eulerpfad vorliegt? Es darf eine Bahnstrecke verändert oder hinzugefügt werden (egal ob Regio oder ICE). (**1 Punkt**)

c) Wie viele Bahnstrecken müssten mindestens hinzugefügt werden, damit ein Eulerkreis vorliegt? (**1 Punkt**)

Aufgabe 4.2: Hamiltonpfade (2 Punkte)

Hamiltonpfade und Hamiltonkreise

Bei großen und komplizierten Graphen haben selbst Computerprogramme sehr schnell Probleme zu überprüfen, ob ein Hamiltonpfad vorliegt. (In der Informatik spricht man auch von einem NP-vollständigen Problem).
Oft lohnt es sich aber, den Graphen, wenn möglich, im Vorhinein zu betrachten. Manchmal fällt einem dort schon ein Problem auf, bevor man anfängt zu rechnen.

a) Warum kann in unserem Ursprungsgraph aus Aufgabe 1 kein Hamiltonpfad oder -kreis existieren? (**2 Punkte**)

Allgemeine Hinweise
Einsendeschluss: Sonntag, 30. November 2025, 19:59 Uhr

Die Abgabe der Lösungen erfolgt über Stud.IP: `https://studip.uni-hannover.de`

Das zulässige Dateiformat für die zusammengeschriebene Lösung (mit eingebetteten Bildern) ist PDF. Gebt auf der ersten Seite eurer PDF-Datei euren Teamnamen, die Namen der einzelnen Teammitglieder sowie deren Schulen an. Benennt eure Datei nach folgendem Schema: „Teamname_Aufgabe2“.

Es ist ausreichend, wenn eine*r von euch die PDF-Datei stellvertretend für das gesamte Team hochlädt. Das Hochladen der Datei funktioniert wie folgt:

Loggt euch mit den bei eurer Anmeldung zur 4 Science Challenge angelegten Zugangsdaten auf der Stud.IP-Seite ein (bitte nutzt dazu den „Login ohne WebSSO“). Geht dann auf „Meine Veranstaltungen“ und auf die 4 Science Challenge 2025/2026. Geht dann oben auf „Dateien“ und auf den Ordner „Upload Aufgabe 2“. Dort könnt ihr entweder über „Dokument hinzufügen“ oder über „Dateien hochladen“ eure Lösungsdatei hochladen.

Wenn ihr die Datei hochgeladen habt, öffnet sich ein Fenster, in dem u. a. nach Lizenzinformationen gefragt wird. Dieses braucht ihr nicht weiter zu beachten und könnt einfach auf „Speichern“ klicken. **Bitte achtet darauf, dass ihr eure Dateien wirklich innerhalb des Ordners „Upload Aufgabe 2“ hochladet und nicht außerhalb davon - andernfalls können die anderen Teams eure Lösung sehen!**

Die Teilnahmebedingungen und weitere Informationen findet ihr unter `www.uni-hannover.de/4sciencechallenge`

Der Rechtsweg ist ausgeschlossen.

Anhang: Beispiel für den Dijkstra-Algorithmus anhand der Strecke Hamburg → München mit vollständiger Distanztabelle

Legende

- Grün: Knoten wurde in dieser Iteration besucht (markiert als fertig)
- Rosa: Distanz wurde in dieser Iteration aktualisiert
- Rot: Zielknoten München wurde erreicht
- Werte in Klammern: (Vorgängerknoten)

Vollständige Distanztabelle

Iter	Besucht	AC	B	HB	DD	EF	F	GÖ	HH	H	K	MD	M	MS	OS	S
0	—	∞	∞	∞	∞	∞	∞	∞	0	∞	∞	∞	∞	∞	∞	∞
1	HH d=0	∞	∞	95 (HH)	∞	∞	∞	∞	0	133 (HH)	∞	∞	∞	∞	∞	∞
2	HB d=95	∞	∞	95	∞	∞	∞	∞	0	133	∞	∞	∞	∞	198 (Br)	∞
3	H d=133	∞	383 (H)	95	∞	∞	∞	227 (H)	0	133	383 (H)	268 (H)	∞	∞	198	∞
4	OS d=198	∞	383	95	∞	∞	∞	227	0	133	383	268	∞	243 (OS)	198	∞
5	GÖ d=227	∞	383	95	∞	367 (GÖ)	407 (GÖ)	227	0	133	383	268	∞	243	198	∞
6	MS d=243	∞	383	95	∞	367	407	227	0	133	373 (MS)	268	∞	243	198	∞
7	MD d=268	∞	383	95	∞	367	407	227	0	133	373	268	∞	243	198	∞
8	EF d=367	∞	383	95	567 (EF)	367	407	227	0	133	373	268	687 **(EF)**	243	198	∞

Abkürzungen: AC = Aachen, B = Berlin, HB = Bremen, DD = Dresden, EF = Erfurt, F = Frankfurt, GÖ = Göttingen, HH = Hamburg, H = Hannover, K = Köln, MD = Magdeburg, M = München, MS = Münster, OS = Osnabrück, S = Stuttgart

Detaillierte Erklärung der obigen Tabelle

Iteration 0: Initialisierung

- Startknoten Hamburg: Distanz = 0
- Alle anderen Knoten: Distanz = ∞

Iteration 1: Besucht Hamburg (d = 0)

Aktualisierungen:

- Bremen: $0 + 95 = 95$ km (via Hamburg, Regio)
- Hannover: $0 + 133 = 133$ km (via Hamburg, ICE)

Nächster Knoten: Bremen (kleinste Distanz = 95)

Iteration 2: Besucht Bremen (d = 95)

Aktualisierungen:

- Hamburg: bereits besucht → überspringen
- Hannover: $95 + 100 = 195$ km > 133 km → keine Verbesserung
- Osnabrück: $95 + 103 = 198$ km (via Bremen, Regio)

Nächster Knoten: Hannover (kleinste Distanz = 133)

Iteration 3: Besucht Hannover (d = 133)

Aktualisierungen:

- Göttingen: $133 + 94 = 227$ km (via Hannover, ICE)
- Köln: $133 + 250 = 383$ km (via Hannover, ICE)
- Berlin: $133 + 250 = 383$ km (via Hannover, ICE)
- Magdeburg: $133 + 135 = 268$ km (via Hannover, Regio)
- Bremen: $133 + 100 = 233$ km > 95 km → keine Verbesserung

Nächster Knoten: Osnabrück (kleinste Distanz = 198)

Iteration 4: Besucht Osnabrück (d = 198)

Aktualisierungen:

- Münster: $198 + 45 = 243$ km (via Osnabrück, Regio)
- Bremen: bereits besucht → überspringen

Nächster Knoten: Göttingen (kleinste Distanz = 227)

Iteration 5: Besucht Göttingen (d = 227)

Aktualisierungen:

- Frankfurt: $227 + 180 = 407$ km (via Göttingen, ICE)
- Erfurt: $227 + 140 = 367$ km (via Göttingen, Regio) ← *Wichtig!*
- Hannover: bereits besucht → überspringen

Nächster Knoten: Münster (kleinste Distanz = 243)

Iteration 6: Besucht Münster (d = 243)

Aktualisierungen:

- Köln: $243 + 130 = 373$ km < 383 km → Verbesserung! (via Münster, Regio)
- Osnabrück: bereits besucht → überspringen

Nächster Knoten: Magdeburg (kleinste Distanz = 268)

Iteration 7: Besucht Magdeburg (d = 268)

Aktualisierungen:

- Hannover: bereits besucht → überspringen
- Erfurt: $268 + 150 = 418$ km > 367 km → keine Verbesserung (Weg über Göttingen ist besser!)

Nächster Knoten: Erfurt (kleinste Distanz = 367)

Iteration 8: Besucht Erfurt (d = 367) — ZIEL ERREICHT!

Aktualisierungen:

- **München:** $367 + 320 = 687$ **km (via Erfurt, ICE) ← ZIELKNOTEN ERREICHT!**
- Dresden: $367 + 200 = 567$ km (via Erfurt, ICE)
- Göttingen: bereits besucht → überspringen
- Magdeburg: bereits besucht → überspringen

Algorithmus stoppt, da München erreicht wurde!

*

Rekonstruktion des kürzesten Pfades
Durch Zurückverfolgen der Vorgängerknoten erhalten wir:

Von	Nach	Distanz	Typ
Hamburg	Hannover	133 km	ICE
Hannover	Göttingen	94 km	ICE
Göttingen	Erfurt	140 km	Regio
Erfurt	München	320 km	ICE
Gesamt		**687 km**	

*

Wichtige Erkenntnisse

1. Der Algorithmus besucht Knoten in **aufsteigender Reihenfolge ihrer Distanz** vom Startknoten
2. Die Route kombiniert **ICE und Regio**-Verbindungen optimal (in Bezug auf das jeweilige Gewicht, hier Distanz)
3. Der Weg über Göttingen → Erfurt (Regio, 140 km) ist kürzer als über Magdeburg → Erfurt (Regio, 150 km)
4. Köln wird zweimal aktualisiert:
   - Iteration 3: 383 km (via Hannover, ICE)
   - Iteration 6: 373 km (via Münster, Regio) — Verbesserung!
5. Nach 8 Iterationen wurde das Ziel erreicht (von 15 Knoten wurden 8 besucht)

*

Warum ist diese Route optimal?
Alternative Routen sind länger:

- **Via Berlin:** Hamburg → H → Berlin → Dresden → Erfurt → München = 1068 km (+381 km)
- **Via Magdeburg:** Hamburg → H → Magdeburg → Erfurt → München = 738 km (+51 km)
- **Via Frankfurt:** Hamburg → H → GÖ → Frankfurt → ... (keine Verbindung nach München)