Lehrstuhl für Informatik I
Algorithmen und Komplexität
Universität Würzburg

Würzburg, den 22. April 2026

Prof. Dr. Alexander Wolff
Antonio Lauerbach, M. Sc.

2. Übungsblatt zur Vorlesung Algorithmische Graphentheorie (Sommer 2026)

Aufgabe 1 – Kleinste Knotenüberdeckung

Das Problem KLEINSTE KNOTENÜBERDECKUNG ist wiefolgt: Gegeben ein ungerichteter Graph $G = (V, E)$, suchen wir eine Teilmenge $C \subseteq V$ minimaler Größe, sodass für jede Kante $\{u, v\} \in E$ mindestens einer der Endpunkte in C enthalten ist, also $u \in C$ oder $v \in C$ gilt.

a) Formulieren Sie das Problem KLEINSTE KNOTENÜBERDECKUNG als ganzzahliges lineares Programm. Argumentieren Sie, dass Ihre Formulierung korrekt ist. Erklären Sie insbesondere die Bedeutung Ihrer Variablen, Zielfunktion und Nebenbedingungen. **3 Punkte**

PuLP b) Implementieren Sie Ihr ganzzahliges lineares Programm mit PuLP. Ihr Programm muss in der Lage sein, das Problem für *beliebige* einfache ungerichtete Graphen zu lösen.

Entwerfen Sie dafür zunächst ein geeignetes Format, mit dem Graphen als Eingabe an Ihr Programm übergeben werden können.

Testen Sie Ihr Programm mit dem Graphen aus Abb. 1 und geben Sie die minimale Knotenüberdeckung an, die Ihr Programm zurückgibt. Laden Sie Ihren Quelltext in WueCampus hoch. **4 Punkte**

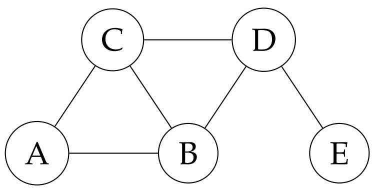


ABBILDUNG 1

Aufgabe 2 – Straßenreparatur mittels Linearer Programmierung und Flüssen

Gegeben sei ein ungerichteter Graph $G = (V, E)$, der ein Verkehrsnetzwerk darstellt, wobei jeder Knoten $v \in V$ eine Stadt repräsentiert und jede Kante $e = \{u, v\}$ eine Straße zwischen den Städten u und v.

Die Straßen müssen erneuert werden. Dabei sind für jede Straße $e \in E$ Reparaturkosten nötig, die durch eine Funktion $r\colon E \to \mathbb{R}_{\geq 0}$ gegeben sind. Zusätzlich hat jede Stadt $v \in V$ nur ein begrenztes Budget, das durch eine Funktion $B\colon V \to \mathbb{R}_{\geq 0}$ gegeben ist.

Hinweis: Sie können hier Python nutzen, um Ihre linearen Programme auf einer kleinen selbst erdachten Instanz zu testen – müssen das aber nicht tun. Wir fordern hier explizit eine mathematische Beschreibung Ihres linearen Programms (für eine unbekannte Instanz), keinen Code. Beschreiben Sie zunächst Variablen und Constraints in Worten.

a) Für jede Straße $e = \{u, v\} \in E$ dürfen sich die beiden Städte u und v die Kosten $r(e)$ in beliebigem Verhältnis teilen, wobei natürlich keine negativen Zahlungen möglich sind. Wir wollen entscheiden, ob es eine zulässige Verteilung der Zahlungen gibt, so dass alle Erneuerungen bezahlt werden, aber keine Stadt ihr Budget überschreitet.

   Geben Sie ein lineares Programm zum Lösen dieses Problems an. Argumentieren Sie, dass Ihre Lösung korrekt ist. **3 Punkte**

b) Wir wollen erneut die Reparaturkosten verteilen, lassen jetzt aber keine Aufteilung von Einzelkosten auf die Städte mehr zu; das heißt, für jede Straße $e = \{u, v\} \in E$ muss jetzt entweder die Stadt u oder die Stadt v den Gesamtbetrag $r(e)$ zahlen. Wir wollen wieder entscheiden, ob es eine zulässige Verteilung unter dieser Zusatzbedingung gibt.

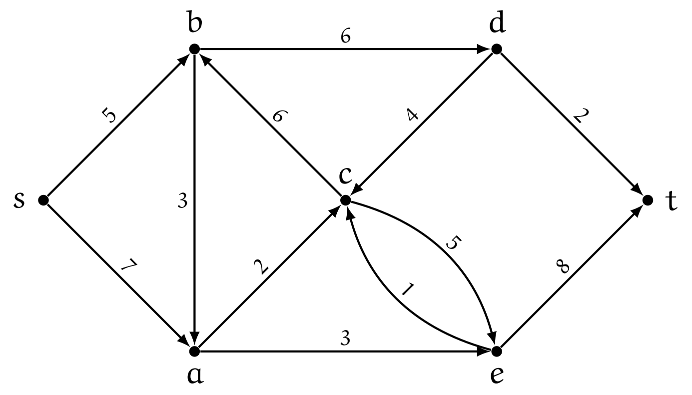
   Lösen Sie dieses Problem mit Hilfe von *ganzzahliger* linearer Programmierung. Argumentieren Sie, dass Ihre Lösung korrekt ist. **2 Punkte**

c) Lösen Sie das Problem aus Teilaufgabe a) erneut, verwenden Sie jetzt aber keine lineare Programmierung, sondern eine Modellierung als Flussnetzwerk. Geben Sie also an, was Ihre Knoten, Kanten, Kantenkapazitäten, Ihr Startknoten s und Ihr Zielknoten t sind. Argumentieren Sie wieder, dass Ihre Lösung korrekt ist. **4 Punkte**

   Hinweis: Argumentieren Sie, wie aus der Lösung für Teilaufgabe a) ein Fluss in Ihrem Netzwerk abgeleitet werden kann und umgekehrt.

Aufgabe 3 – Flüsse finden

Gegeben sei der folgende Graph mit Kantenkapazitäten:



a) Ermitteln Sie einen maximalen s-t-Fluss. Wie groß ist der Wert des Flusses?
4 Punkte

b) Begründen Sie, warum es keinen Fluss mit größerem Flusswert geben kann.
1 Zusatzpunkt

Bitte geben Sie Ihre Lösungen bis **Dienstag, 5. Mai 2026, 13:00 Uhr** einmal pro Gruppe über WueCampus als PDF-Datei ab. Geben Sie stets die Namen aller an, die das Übungsblatt bearbeitet haben (max. 2).

Begründen Sie Ihre Behauptungen und kommentieren Sie Ihren Pseudocode!

Aufgaben, die mit PuLP gekennzeichnet sind, fordern das Erstellen und Lösen von linearen Programmen. Geben Sie Ihren kommentierten zusammen mit Ihrer Bearbeitung des Übungsblatts auf WueCampus ab. Der Quellcode muss von derselben Person abgegeben werden, die auch das PDF hochgeladen hat.

Plagiate werden mit 0 Punkten für das ganze Übungsblatt gewertet.