Lehrstuhl für Informatik I
Optimierung unter Ressourcenbeschränkung
Universität Würzburg

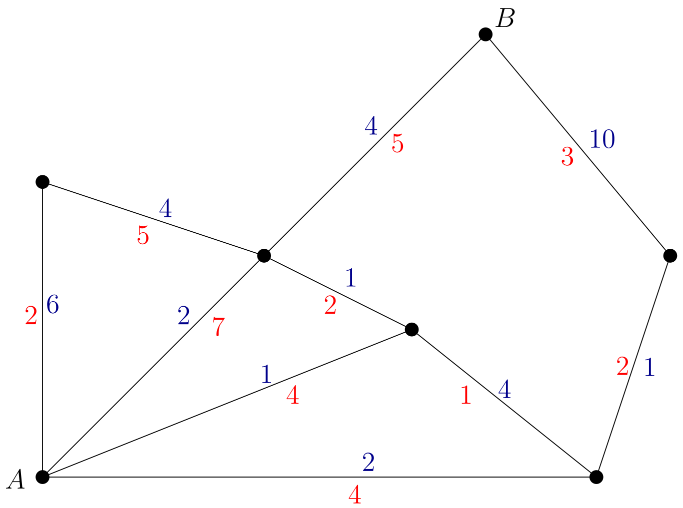
Prof. Dr. Marie Schmidt
Kendra Reiter, M. Sc.
SS 2023

2. Übungsblatt zur Vorlesung Graphen und diskrete Optimierung

Abgabe am 3.05.2023 um 10:00 Uhr

Aufgabe 1. *(Multikriterielle Wege)* **7 Punkte**

Durch den folgenden Graphen wird ein Wegenetz mit verschiedenen Straßen (Kanten) und Haltestellen (Ecken) dargestellt.



Jede Kante des Graphs hat zwei Kantengewichte: die blaue Zahl steht für die erwartete Fahrtzeit und die rote Zahl für die erwarteten Spritverbrauch auf der Kante. Wir wollen mithilfe des in der Vorlesung vorgestellten *Label-Setting Algorithmus* alle nichtdominierten Wege von A nach B finden.

(1.1) Ergänzen Sie den Pseudocode für den Label-Setting Algorithmus aus der Vorlesung um eine Auswahlfunktion für die Label aus der Menge L_{open} und um eine *Dominanzcheck*-Funktion. **2 Punkte**

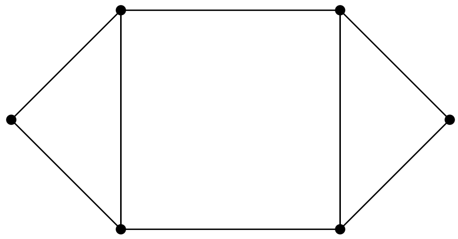
(1.2) Nutzen Sie den so ergänzten Algorithmus dazu, im obrigen Graph die Menge der nichtdominierten Wege von A nach B zu finden. **1 Punkt**

(1.3) Bestimmen Sie die Laufzeit: **4 Punkte**

- (a) Beweisen Sie, dass die Laufzeit des Algorithmus exponentiell in der Größe des Input sein kann. Konstruieren Sie dazu einen Graph, in dem die Anzahl der nichtdominierten Wege (also der Output) schon exponentiell in der Größe des Input ist.
- (b) Wegen des in (a) zu konstruierenden Beispiels ist es in diesem Fall sinnvoller, die worst-case Laufzeit des Algorithmus' als Funktion von Inputlänge n *und* Outputlänge m anzugeben. Können Sie zeigen, dass die worst-case Laufzeit des Algorithmus polynomiell in $n + m$ ist? Oder, im Gegenteil, beweisen dass die Laufzeit auch in dieser Größe im schlechtesten Fall exponentiell ist?

Aufgabe 2. *(Eulertouren)* **4 Punkte**

Sei $G = (V, E)$ der folgende Graph:



(2.1) Wie viele (und welche) Kanten muss man zu G hinzufügen, damit G eine Eulertour enthält? **1 Punkt**

(2.2) Beweisen Sie, oder geben Sie ein Gegenbeispiel mit Begründung an: gegeben sei ein beliebiger zusammenhängender Graphen, der nicht schon selber eine Eulertour enthält. Kann man zu diesem Graphen genau einen Knoten hinzufügen, und passende Kanten an diesen Knoten hinzufügen, sodass es eine Eulertour in dem neuen Graphen gibt? **3 Punkte**

Aufgabe 3. *(Existzenz eines Hamiltonkreis)* **3 Punkte**

Sei $G = (V, E)$ ein einfacher zusammenhängender Graph, in dem es zwei nicht benachbarte Knoten x und y gibt, für die $\deg(x) + \deg(y) \geq |V|$ gilt.
Beweisen Sie: G besitzt genau dann einen Hamiltonkreis, wenn der Graph $G' = (V, E \cup \{\{x, y\}\})$ einen Hamiltonkreis besitzt.

Aufgabe 4. *(Programmieraufgabe)* **6 Punkte**

Implementieren Sie Cheapest-Insertion für das Handelsreisendenproblem in Java. Den Pseudocode für Cheapest-Insertion finden Sie auf den Folien der zweiten Vorlesung.

(a) Implementieren Sie hierfür ein Objekt, mit dem Sie einen gewichteten Graphen als Eingabe für Ihren Algorithmus definieren können. Ihr Algorithmus sollte die gefundene optimale Tour auf diesem Graphen zurückgeben.

(b) Geben Sie die (worst-case) Laufzeit Ihrer Algorithmen in den Eingabegrößen $|V|$ und $|E|$ an.

(c) Konstruieren Sie ein Beispiel, für welches Ihr Algorithmus nicht die optimale Tour ausgibt. Welche Ausgabe hat Ihr Programm und was wäre die optimale Tour?

Laden Sie für diese Aufgabe bitte zusätzlich zu der pdf-Abgabe der Lösungen Ihren Quellcode auf WueCampus hoch.

Bitte laden Sie Ihre Lösungen als pdf Datei auf WueCampus hoch. Die Übungsblätter können in Gruppen von maximal zwei Personen bearbeitet werden, geben Sie bei der Abgabe stets beide Namen an.