Karlsruher Institut für Technologie
Institut für Theoretische Informatik

Prof. Dr. Dennis Hofheinz
Lukas Barth, Lisa Kohl

11. Übungsblatt zu Algorithmen I im SoSe 2016

https://crypto.iti.kit.edu/index.php?id=algo-sose16
{lisa.kohl,lukas.barth}@kit.edu

Aufgabe 1 (*Lineare Programmierung, 4 Punkte*)

Dem ebenso genialen wie empfindlichen Superbösewicht Doktor Meta ist zu Ohren gekommen, dass ihn nicht alle Studenten mögen. Nun möchte er herausfinden, um wen es sich dabei handelt. Er hat dazu nicht nur Røbøt, sondern mittlerweile ein ganzes Heer von insgesamt I ebenso folgsamen wie fehlkonstruierten Robotern R_i zur Verfügung. Nach einer Anzahl von Tests hat er für J verschiedene Arten von Arbeit A_j herausgefunden, wieviel es bringt, Roboter R_i die Arbeit A_j eine Stunde lang ausführen zu lassen (dieser Wert sei mit a_{ij} bezeichnet). Die Roboter laufen im Batteriebetrieb und müssen nach vier Stunden wieder aufgeladen werden. Doktor Meta möchte nun herausfinden, wie er die zu erledigenden Arbeiten optimal an seine Roboter verteilen kann, sodass der Wertgewinn maximal ist. Zwischendurch soll dabei kein Akku aufgeladen werden und am Ende soll x_{ij} (für $1 \leq i \leq I$ und $1 \leq j \leq J$) dem Anteil von R_i's Zeit an der Aufgabe A_j entsprechen. Die Aufgabe A_j ist erledigt, sobald z_j Stunden an ihr gearbeitet wurde (egal von welchem Roboter), danach bringt das Weiterarbeiten an dieser Aufgabe keinen zusätzlichen Wertgewinn.
Beispielsweise sei für $R_1 =$ Røbøt der Wert für die Aufgabe A_1 „finde einen Weg" mit $a_{11} = 0,1$ gering und die Aufgabe A_2 „führe einen Algorithmus aus" mit $a_{12} = 10$ vergleichsweise groß. Røbøt eine halbe Stunde daran arbeiten zu lassen Wege zu finden und eine halbe Stunde daran Algorithmen auszuführen (also $x_{11} = x_{12} = 0,5$) würde dementsprechend nur einen Wertgewinn von $0,1 \cdot 0,5 + 10 \cdot 0,5 = 5,05$ bringen, Røbøt stattdessen eine Stunde Algorithmen ausführen zu lassen dagegen einen Gewinn von 10, hierbei $z_1 \geq 0,5$ und $z_2 \geq 1$ angenommen.
Stellen Sie ein geeignetes Lineares Programm auf, dass dieses Problem formalisiert und Doktor Meta hilft, auch den letzten Studenten von seiner Liebenswürdigkeit zu überzeugen. Überlegen Sie dabei zunächst genau, welche Nebenbedingungen (Constraints) für x_{ij} gelten müssen. Wieviele gibt es insgesamt? Erklären Sie jeweils die Bedeutung der aufgestellten (Un-)Gleichungen.

Aufgabe 2 (Vertex Cover, $2 + 5 + 1$ *Punkte und bis zu* 3 *Bonuspunkte*)

Wir betrachten das in der Übung vorgestellte Problem Vertex Cover:

Gegeben ist ein ungerichteter Graph $G = (V, E)$.

Gesucht ist eine minimale Menge $S \subseteq V$, sodass $\forall \{u, v\} \in E \colon \{u, v\} \cap S \neq \emptyset$.

a) Betrachten Sie den in der Übung vorgestellten Algorithmus Approx-Vertex-Cover. Geben Sie einen Graphen an, bei dem dieser Algorithmus *immer* eine nicht-optimale Lösung ausgibt.

b) Entwerfen Sie einen effizienten Greedy-Algorithmus, der des Vertex Cover-Problem auf Bäumen in Linearzeit optimal löst. Begründen Sie, warum Ihr Algorithmus funktioniert.
Hinweis: Für Algorithmen, die eine schlechtere Zeitkomplexität aufweisen, werden unter Umständen Teilpunkte vergeben.

c) Der leicht verwirrte Professor Peter R. Oblem schlägt folgenden Algorithmus zur Lösung des Vertex Cover-Problems vor: Wähle immer den Knoten mit höchstem Grad (d.h. der größten Anzahl von inzidenten Kanten) aus, füge ihn zum Vertex Cover hinzu, und lösche ihn samt aller seiner Kanten. Wiederhole, bis der Graph leer ist.

c1) Zeigen Sie (z.B. mittels eines Beispiels), dass der vorgeschlagene Algorithmus nicht optimal ist.
Hinweis: Diese Teilaufgabe gibt einen Punkt.

c2) Zeigen Sie, dass es sich bei dem Algorithmus auch nicht um eine $1,5$-Approximation handelt, d.h. dass es Fälle gibt, in denen das berechnete Vertex Cover um mehr als einen Faktor $1,5$ größer ist als das optimale Vertex Cover.
Hinweis: Diese Teilaufgabe gibt 2 Bonuspunkte.

c3) Zeigen Sie, dass es sich bei dem Algorithmus nicht einmal um eine 2-Approximation handelt, d.h. dass es Fälle gibt, in denen das berechnete Vertex Cover um mehr als einen Faktor 2 größer ist als das optimale Vertex Cover.
Hinweis: Diese Teilaufgabe löst offensichtlich auch Teilaufgabe c2, und gibt *zusätzlich zu c2)* einen weiteren Bonuspunkt.

Aufgabe 3 (*Steinerbäume - Güte von Approximationsalgorithmen,* 7 *Punkte*)

Wir betrachten den in der Übung vorgestellten Approximationsalgorithmus zur Konstruktion von Steinerbäumen (10. Übung, Folie 67).

a) Geben Sie einen geeigneten ungerichteten zusammenhängender Graph $G = (V, E)$ und eine Menge $S \subseteq V$ mit $|S| = 4$ an für den das Folgende gilt: Es gibt einen Steinerbaum, der mit Hilfe des Algorithmus aus der Vorlesung konstruiert wurde und dessen Gesamtgewicht **genau** dem $1,5$-fachen des Gesamtgewichts eines optimalen Steinerbaums entspricht. Geben Sie den konstruierten und einen optimalen Steinerbaum an.

b) Sei $T = (V, E)$ ein ungerichteter Baum mit $\mathbf{m} = |\mathbf{E}|$ Kanten. Zeigen Sie: Es gibt einen geschlossenen Pfad $P = \langle v_0, v_1, \ldots, v_{2m} \rangle$ mit $v_{2m} = v_0$ in T, für den gilt

   i.) jede Kante von T wird von P genau zweimal durchlaufen,

   ii.) jedes Blatt von T kommt genau einmal in P vor (wobei der Anfangs- bzw. Endpunkt $v_0 = v_{2m}$ als ein Vorkommen in P zählt) und

   iii.) wenn v_i und v_j zwei Blätter in P sind, zwischen denen kein anderes Blatt in P vorkommt, dann ist der Teilpfad $v_i, v_{i+1}, \ldots, v_j$ ein einfacher Pfad (d.h. jede Kante wird höchstens einmal durchlaufen).

c) Sei $G = (V, E)$ ein ungerichter, zusammenhängender Graph und c_{opt} das Gewicht eines optimalen Steinerbaums für $S \subseteq V$. Zeigen Sie, dass der in der Übung vorgestellte Algorithmus einen Steinerbaum mit Gewicht c_{alg} liefert, sodass $c_{\text{alg}} \leq 2 \cdot \left(1 - \frac{1}{|S|}\right) \cdot c_{\text{opt}}$ gilt.

Ausgabe: Mittwoch, 06.07.2016
Abgabe: Freitag, 15.07.2016, 12:45 im Briefkasten im Untergeschoss von Gebäude 50.34

Deckblatt Übungsblatt 11
Algorithmen I

A1	A2	A3	$\sum$

Tutoriumsnummer:

Name	Matrikelnummer	Unterschrift

Mit unseren Unterschriften bestätigen wir, dass die Aufgaben von den Unterzeichnern eigenständig gelöst worden sind.

Hinweis: Das Übungsblatt darf in Gruppen von bis zu zwei Personen bearbeitet werden. Beide Personen müssen demselben Tutorium zugeteilt sein. Möchte jemand seine Abgaben-Gruppe innerhalb des Semesters wechseln, so ist dies im Voraus mit dem Tutor abzusprechen. **Bitte tragen Sie in das obere Quadrat *groß* die Nummer Ihres Tutoriums ein.** Die Lösung des Übungsblattes ist in jedem Fall mit diesem Deckblatt abzugeben.