Institut für Theoretische Informatik
Lehrstuhl Prof. Dr. D. Wagner

Übungsblatt 6

Vorlesung Theoretische Grundlagen der Informatik im WS 16/17

Ausgabe 22. Dezember 2016
Abgabe 17. Januar 2017, 11:00 Uhr (im Kasten im UG von Gebäude 50.34)

Bitte nutzen Sie den WebInScribe Deckblattgenerator
und heften Sie das Deckblatt an Ihr Übungsblatt.
`https://webinscribe.ira.uka.de/deckblatt/index.php?course=10588`.

Aufgabe 1

(2 + 2 = 4 Punkte)

Das Bin Packing-Problem ist wie folgt definiert:

Gegeben: endliche Menge $M = \{a_1, \ldots, a_n\}$ mit Gewichtungsfunktion $s : M \to (0, 1]$.
Gesucht: Minimale Anzahl an Bins m, so dass für jeden Bin B_i mit $i = 1, \ldots, m$ gilt:

$$\sum_{a \in B_i} s(a) \leq 1$$

In der Übung wurde ein Approximationsalgorithmus zusammen mit einer oberen Schranke von 2 für dessen relative Güte vorgestellt.

(a) Zeigen Sie, dass diese obere Schranke gewissermaßen optimal ist. Geben Sie dazu für jedes $1 \geq \epsilon > 0$ eine Folge von Elementen an, so dass der Approximationsalgorithmus bei Abarbeitung dieser Folge mindestens $2(1 - \epsilon)$ Mal so viele Bins füllt, wie ein optimaler Algorithmus.

(b) Die Strategie, um Elemente in die Bins einzufügen, wird nun verändert. Statt nur den letzten Bin zu betrachten, wird jetzt ein Element in den ersten Bin eingefügt, in dem noch ausreichend Platz ist. Zeigen Sie, dass für diesen neuen Approximationsalgorithmus $\mathcal{A}$ gilt $\mathcal{R}_{\mathcal{A}} > \frac{11}{7}$.

Aufgabe 2

(1 + 1 = 2 Punkte)

(a) Sei Π ein $\mathcal{NP}$-vollständiges Problem zu dem ein pseudo-polynomialer Algorithmus existiert. Warum impliziert dies nicht die Existenz eines pseudo-polynomialen Algorithmus für jedes $\mathcal{NP}$-vollständige Problem?

(b) Zeigen Sie, dass ein stark $\mathcal{NP}$-vollständiges Problem genau dann von einem pseudo-polynomialen Algorithmus entschieden wird, wenn $\mathcal{P} = \mathcal{NP}$ gilt.

Aufgabe 3 (1 + 2 + 3 = 6 Punkte)

Betrachten Sie den folgenden Algorithmus Simple-Matcher, der für einen bipartiten Graphen $G = (A \cup B, E)$ ein Matching berechnet.

Ein Matching $M \subset E$ ist eine Menge von Kanten, so dass keine zwei Kanten in M inzident sind. Ein Matching ist *inklusion-maximal*, wenn kein Matching M' existiert, dass M als echte Teilmenge enthält. Ein Matching ist *kardinalitäts-maximal*, wenn kein Matching mit größerer Kardinalität existiert.

Algorithmus 1 : Simple-Matcher

Eingabe : Bipartiter Graph $G = (A \cup B, E)$
Ausgabe : Matching M
$M \leftarrow \emptyset$;
Für $u \in B$
  Wenn *u einen Nachbarn v hat, der zu keiner Kante in M inzident ist,*
    füge Kante $\{u, v\}$ zu M hinzu.;
return M

(a) Zeigen Sie mithilfe eines Beispielgraphen $G = (A \cup B, E)$, dass Simple-Matcher ein Matching M liefern kann, das nur halb so groß ist, wie ein kardinalitäts-maximale Matching $M^\star$ in G. Begründen Sie Ihre Antwort, indem Sie ein kardinalitäts-maximales Matching $M^\star$ angeben, Simple-Matcher auf G ausführen und in jedem Schritt M und den bearbeiteten Knoten angeben.

(b) Sei M ein Matching, das von Simple-Matcher berechnet wurde. Zeigen Sie, dass M ein inklusions-maximales Matching ist.

(c) Zeigen Sie, dass SIMPLE-MATCHER ein Approximationsalgorithmus mit relativer Gütegarantie 2 für das Problem ein maximales Matching in einem bipartiten Graphen zu berechnen ist.

Aufgabe 4

(2 + 2 + 3 = 7 Punkte)

Algorithmus 2 : MAXSUM-SCHEMA

Eingabe : $L \subset \mathbb{N}_0$, Zahl $k \in \mathbb{N}$, $\delta \in \mathbb{R}_{\geq 0}$

$S = \{0\}$;
Für $x \in L$
    $T \leftarrow \{x + y \mid y \in S\}$;
    $X \leftarrow S \cup T$;
    $s \leftarrow$kleinstes Element von X;
    $S \leftarrow \{s\}$;
    Für $y \in X$ *in aufsteigender Reihenfolge*
        Wenn $s \cdot (1 + \delta) < y \leq k$
            $S \leftarrow S \cup \{y\}$;
            $s \leftarrow y$;
return $\max_{y^\star \in S} y^\star$

(a) Bezeichne mit Σ_X die Summe $\Sigma_{x \in X} x$ für eine Menge x. Zeigen Sie, dass für $\delta = 0$ der Algorithmus MAXSUM-SCHEMA die Funktion $\max_{S' \in \{S | S \subset L \wedge \Sigma_S \leq k\}} \Sigma_{S'}$ berechnet. Analysieren Sie die Laufzeit.

(b) Bezeichne mit Y_i die Menge S nach Iteration i mit $\delta = 0$ und Z_i die entsprechende Menge für ein beliebiges $\delta > 0$. Nehmen Sie für beide Menge $Y_0 = Z_0 = \{0\}$ an. Zeigen Sie, für $y \in Y_i$ existiert ein $z \in Z_i$ mit folgender Eigenschaft:

$$\frac{y}{(1 + \delta)^i} \leq z$$

(c) Zeigen Sie, dass MAXSUM-SCHEMA für ein geeignetes δ ein vollständiges polynomiales Approximationschema ist.

Die Schranke ist polynomial in n, k und $1/\epsilon$. Alle Operationen des Algorithmus können ebenfalls in polynomieller Zeit implementiert werden, womit gezeigt ist, dass der Algorithmus für unser δ tatsächlich ein FPAS ist.

Aufgabe 5

(2 Punkte)

Formulieren Sie das Problem EXACT COVER als INTEGER LINEAR PROGRAM. Begründen Sie Ihre Modelierung.

Aufgabe 6 (4 Punkte)

Der ebenso geniale wie auch skrupellose Wissenschaftler und Superbösewicht Doktor Meta ist in Weihnachtsstimmung. Entsprechend groß sind seine Pläne für Elsas Geschenk. Nachdem ihm aber selbst die Allianz der Diktatoren die nötige finanzielle Unterstützung verweigert, muss er nun selber Geld verdienen. Ein einfacher Bankraub ist einem Wissenschaftler und Superbösewicht selbstverständlich nicht würdig. Doktor Meta hat eine bessere Idee: er wird den Weihnachtsmann ausrauben! Mit Geschenken für über sieben Milliarden Menschen ist für Metas Zukunft ausgesorgt.

Das Problem an Doktor Metas Plan ist, dass der Weihnachtsmann, wenn er erst einmal auf seiner Route unterwegs ist, schwer zu schnappen ist. Doktor Meta muss deshalb verhindern, dass der Weihnachtsmann überhaupt erst eine Route findet. Sitzt der Weihnachtsmann dann am Nordpol fest, kann Doktor Meta ihn dort, fernab jeglicher Zeugen, ausrauben.

Meta weiß, dass das Traveling Santa-Problem

Gegeben: Graph $G = (V, E)$ vollständig und gewichtet mit Gewichtungsfunktion $c : E \to \mathbb{Q}$. **Gesucht:** Optimale Tour in G bezüglich c.

im Allgemeinen $\mathcal{NP}$-schwer ist und deshalb praktisch nicht gelöst werden kann. Er weiß aber auch, dass das Traveling Santa-Problem *mit* Dreiecksungleichung durch einen polynomialen Algorithmus mit relativer Gütegarantie von 2 approximiert werden kann – eine solche Tour wäre für den Weihnachtsmann immer noch gut genug.

Doktor Metas Blick fällt auf seinen Patentprojektor, mit dem er den euklidischen Raum in den Metaraum projizieren kann, in dem die Dreiecksungleichung nicht gilt. Helfen Sie Doktor Meta, indem Sie zeigen, dass der Weihnachtsmann im Metaraum eine optimale Route nicht einmal mit konstanter relativer Güte approximieren kann.

Zeigen Sie dazu, dass für das Traveling Santa-Problem *ohne* Dreiecksungleichung ein Approximationsalgorithmus mit konstanter relativer Gütegarantie genau dann existiert, wenn $\mathcal{P} = \mathcal{NP}$ gilt. Sie dürfen dabei die $\mathcal{NP}$-Vollständigkeit der folgenden Probleme nutzen:

Induzierter Pfad:

Gegeben: Ungerichteter Graph $G = (V, E)$ und Parameter $k \in \mathbb{N}$. **Gesucht:** Menge $V' \subseteq V$, $\lvert V' \rvert = k$, so dass der durch V' induzierte Subgraph von G ein Pfad ist.

Hamiltonpfad:

Gegeben: Ungerichteter Graph $G = (V, E)$. **Gesucht:** Geschlossener Pfad in G, der jeden Knoten genau einmal enthält.

Längster Pfad:

Gegeben: Ungerichteter Graph $G = (V, E)$. **Gesucht:** Längster Pfad in G.

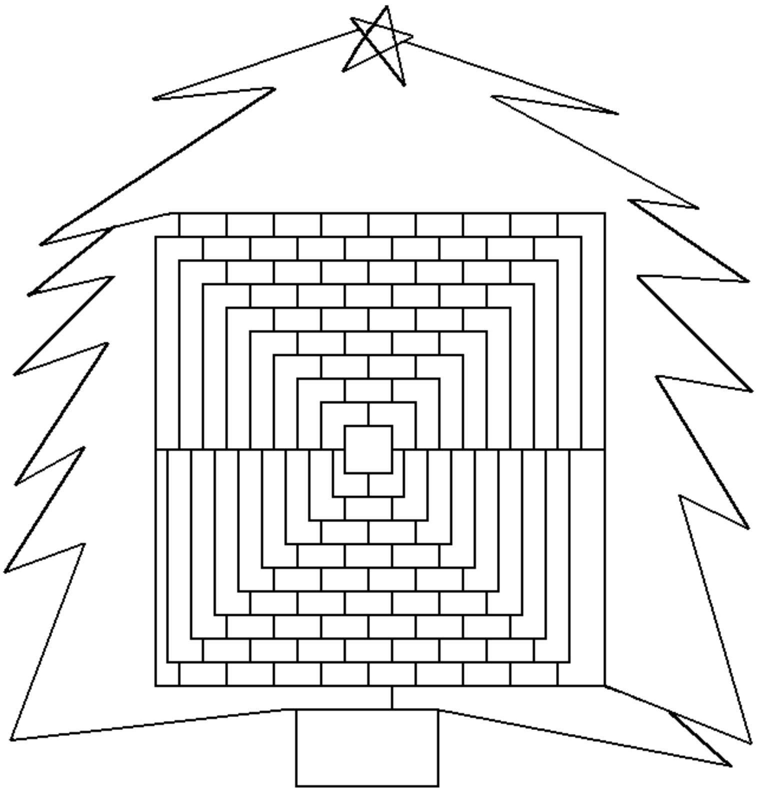
Hinweis: Überlegen Sie sich, wo genau die Dreiecksungleichung im Beweis aus der Vorlesung genutzt wurde. Welches Problem entsteht, wenn die Dreiecksungleichung nicht mehr gilt?

Aufgabe 7 (2 Punkte)

Ein *planarer Graph* ist ein Graph, der so in der Ebene gezeichnet werden kann, dass sich keine zwei Kanten kreuzen. Die *Facetten* eines planaren Graphen bezüglich einer gegebenen Einbettung sind die 'maximalen, durch Kanten abgeschlossenen Flächen'. Insbesondere wird das 'den Graphen umgebende Gebiet' als *Äußere Facette* bezeichnet. Zwei Facetten sind *adjazent*, falls sie durch eine gemeinsame Kante begrenzt werden.

Die nachfolgende Zeichnung ist als planarer Graph zu betrachten, wobei die Knoten implizit als Schnitt- bzw. Berührungspunkte der Kanten gegeben seien.

Färben Sie die Facetten des Graphen mit vier Farben so, dass keine zwei adjazenten Facetten dieselbe Farbe haben.



FROHE WEIHNACHTEN, einen BRAUSENDEN JAHRESWECHSEL
und ein ERFOLGREICHES JAHR 2017