Prof. Dr. Thomas Schneider SoSe 2018

Komplexitätstheorie

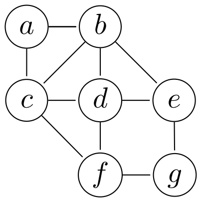
Übungsblatt 1

Abgabe im PDF-Format bis 17. 4. 2018, 23:59 Uhr in Stud.IP, Ordner „Abgabe Übungsblatt 1“
Bitte nur eine PDF-Datei pro Gruppe, Lizenz „Selbst verfasstes, nicht publiziertes Werk“.

---

1. (20 %) Betrachte den Algorithmus `b-clique` aus Kapitel 1 der Vorlesung.

a) Wende den Algorithmus auf folgenden Graphen G und Cliquengröße 3 an:



b) Zeige durch Angeben eines Gegenbeispiels, dass das Eliminieren nicht adjazenter Knoten notwendig ist, weil der Algorithmus sonst nicht korrekt ist.

2. (15 %) Das Rucksackproblem ist als Optimierungsproblem wie folgt formuliert, wobei alle Zahlen *binär kodiert* sind:

Gegeben:

- eine Menge $A = \{a_1, \dots, a_n\}$ von Gegenständen, und für jeden Gegenstand a_i ein Gewicht $g_i \in \mathbb{N}$ und einen Nutzen $n_i \in \mathbb{N}$ sowie
- eine Gewichtsgrenze $G \in \mathbb{N}$ für einen zu packenden Rucksack

Ausgabe: nutzenmaximale Rucksackfüllung, d. h.
Teilmenge $R \subseteq A$, so dass $\sum\limits_{a_i \in R} g_i \leq G$ und $\sum\limits_{a_i \in R} n_i$ maximal

Berechne die Lösung für folgende Eingabe:

$$A = \{a_1, \dots, a_5\} \quad (n_1, \dots, n_5) = (3, 8, 3, 6, 2) \quad (g_1, \dots, g_5) = (2, 6, 2, 4, 3) \quad G = 8$$

Argumentiere, dass die Lösung wirklich optimal ist.

3. (25 %) In der Variante als Berechnungsproblem ist beim Rucksackproblem zusätzlich ein Zielnutzen N gegeben, der mindestens erreicht werden muss. Ausgegeben wird eine Rucksackfüllung, die diesen Zielnutzen realisiert.

Zeige: wenn das Berechnungsproblem in Polynomialzeit lösbar ist, dann auch das Optimierungsproblem.
Hinweis: es ist nicht möglich, alle Werte von 0 bis $\sum\limits_{a_i \in A} n_i$ durchzugehen: die Werte n_i sind binär gegeben.

4. **(20 %)** Um Entscheidungsprobleme für Graphen als formale Sprache definieren zu können, benötigt man eine Kodierung von Graphen als Wörter über einem festen Alphabet Σ. Gib eine Kodierung für gerichtete Graphen an, die auf Adjazenzlisten beruht (siehe Beispiel aus Kapitel 1 der Vorlesung). Gesucht ist also eine Vorschrift, wie man zu einem gegebenen Graphen $G = (V, E)$ ein Wort $w_G \in \Sigma^*$ erhält, das G repräsentiert. Dabei darf jedes Wort $w \in \Sigma^*$ natürlich nur höchstens einen Graphen (bis auf Umbenennung der Knoten) repräsentieren. Des Weiteren darf Σ *nicht* von der Größe des Graphen abhängen.

   Gib außerdem ein Beispiel für einen Graphen G und das zugehörige Wort w_G an.

5. **(20 %)** Entwickle eine (deterministische) Turingmaschine, die als Eingabe $\$\mathsf{bin}(n)$ erhält, wobei $\mathsf{bin}(n)$ die binäre Kodierung der Zahl n ist, und die diese dann inkrementiert. Es wird angenommen, dass das höchstwertige Bit ganz links steht und das niedrigstwertige ganz rechts. Beim Start steht der Kopf der Maschine auf dem Symbol \$. Dieses darf beim Inkrementieren wenn nötig überschrieben werden.

   Gib die Übergänge in graphischer Form an (wie in der Vorlesung). Erkläre die Konstruktion. Gib die Berechnung der TM auf der Eingabe \$111 an.

6. **Zusatzaufgabe (20 %)** Beweise die Korrektheit des Algorithmus `b-clique` aus Kapitel 1 der Vorlesung. Zeige dazu: wenn G eine k-Clique enthält, dann gibt `b-clique`(G, k) eine k-Clique in G aus.

   Hinweis: verwende Induktion über k.

LaTeX-Tipp: Wenn Ihr den Befehl für ein bestimmtes Symbol sucht, könnt Ihr in `symbols-a4.pdf` nachschauen, die in jeder LaTeX-Installation enthalten ist. Die neueste Version findet Ihr auch im Internet: `https://tinyurl.com/symbols-a4-pdf`

Außerdem kann ich die Webapp `Detexify` sehr empfehlen – sie erspart Euch das Durchblättern der PDF-Datei: `https://tinyurl.com/detexify-2018`