

Theorie der Informatik

M. Helmert
F. Pommerening
Frühjahrssemester 2016

Universität Basel
Fachbereich Informatik

Übungsblatt 13

Abgabe: Mittwoch, 1. Juni 2016

Anmerkung: Für Abgaben, die ausschliesslich mit $\text{\LaTeX}$ erstellt wurden, gibt es einen Bonuspunkt. Bitte geben Sie nur die resultierende PDF-Datei bzw. einen Ausdruck davon ab.

Hinweis: Dies ist das letzte reguläre Übungsblatt der Vorlesung. Nächste Woche wird noch ein Übungsblatt ausgegeben, das nur aus Bonusaufgaben besteht und die Kapitel E3–E5 behandelt. Es gibt damit insgesamt ohne Bonuspunkte ein Maximum von 120 Punkten. Für die Prüfungszulassung werden wie angekündigt 50% dieser Punkte, also 60 Punkte, benötigt.

Aufgabe 13.1 (Algorithmen für `CLIQUE`, 1.5+1.5 Punkte + 1 Bonuspunkt)

Betrachten Sie das Entscheidungsproblem `CLIQUE`:

- *Gegeben:* ungerichteter Graph $G = \langle V, E \rangle$, Zahl $K \in \mathbb{N}_0$
 - *Gefragt:* Enthält G eine Clique der Grösse K , also eine Menge von Knoten $C \subseteq V$ mit $|C| \geq K$ und $\{u, v\} \in E$ für alle $u, v \in C$ mit $u \neq v$?
- (a) Geben Sie einen nichtdeterministischen Algorithmus für `CLIQUE` an, dessen Laufzeit durch ein Polynom in $|V| + |E|$ beschränkt ist. Begründen Sie, warum die Laufzeit des Algorithmus polynomiell ist.
- (b) Geben Sie einen deterministischen Algorithmus für `CLIQUE` an.
- (c) *Bonusaufgabe:* Schätzen Sie die Laufzeit Ihres Algorithmus aus Teil (b) in O -Notation ab.

Sie dürfen bei Ihrer Antwort beliebige übliche Programmierkonzepte verwenden. Es ist *nicht* nötig, dass Sie sich auf die eingeschränkte Syntax von `WHILE`-Programmen o.ä. beschränken. Es ist ausreichend high-level Pseudocode anzugeben, solange klar ist, dass jeder einzelne Schritt in polynomieller Zeit ausgeführt werden kann. Verwenden Sie die in der Vorlesung beschriebenen `GUESS`-Anweisungen für nichtdeterministische Anweisungen.

Aufgabe 13.2 (P vs. NP, 1+1+1.5+1.5 Punkte)

Beweisen oder widerlegen Sie die folgenden Aussagen. Geben Sie in jedem Fall einen kurzen Beweis an (2–3 Sätze genügen).

- (a) Sei X ein NP-hartes Problem und Y ein Problem, für das $X \leq_P Y$ gilt. Dann ist Y NP-hart.
- (b) Sei X ein NP-hartes Problem. Wenn für X ein deterministischer polynomieller Algorithmus existiert, dann existiert auch für `DIRHAMILTONCYCLE` ein deterministischer polynomieller Algorithmus.
- (c) Es gibt NP-vollständige Probleme X und Y , so dass für X ein deterministischer polynomieller Algorithmus existiert, aber nicht für Y .
- (d) Sei $Y \subseteq \Sigma^*$ ein beliebiges Problem mit $Y \neq \emptyset$ und $Y \neq \Sigma^*$. Für alle $X \in \text{P}$ gilt $X \leq_P Y$.

Aufgabe 13.3 (Polynomielle Reduktion, 2 Punkte + 1 Bonuspunkt)

Ein *Hamiltonpfad* ist analog zu einem Hamiltonkreis definiert (vgl. Kapitel E1), nur dass ein einfacher Pfad statt einem Kreis gesucht ist. Genauer: ein Hamiltonpfad in einem gerichteten Graphen $\langle V, E \rangle$ ist eine Knotenfolge $\pi = \langle v_1, \dots, v_n \rangle$, die einen Pfad definiert ($\langle v_i, v_{i+1} \rangle \in E$ für alle $1 \leq i < n$) und jeden Knoten des Graphen genau einmal enthält.

Betrachten Sie das Entscheidungsproblem DIRHAMILTONPATH:

- *Gegeben:* gerichteter Graph $G = \langle V, E \rangle$
 - *Gefragt:* Enthält G einen Hamiltonpfad?
- (a) Beweisen Sie, dass DIRHAMILTONPATH NP-hart ist. Sie dürfen dabei verwenden, dass das Problem DIRHAMILTONCYCLE NP-vollständig ist.
- (b) *Bonusaufgabe:* Ist DIRHAMILTONPATH NP-vollständig? Begründen Sie ihre Antwort.