Prof. Dr. M. Chimani
L. Enz, J. Kirstein, T. Oelschlägel, F. Stutzenstein, N. Troost
`https://kreuzerl.tcs.uos.de`
Universität Osnabrück
Theoretische Informatik
Sommersemester 2020

Übungsblatt 9 zur Einführung in die Theoretische Informatik

Ausgabe: 26. Juni 2020 — Kreuzerl-Deadline: 05. Juli 2020

Die Aufgaben auf diesem Blatt beziehen sich auf den Vorlesungsstoff bis inklusive Kapitel 7.8: Arten der Randomisierung.

Aufgabe 9.1 Kein Märchen!

Gegeben das folgende Optimierungsproblem $\mathcal{X}_O$:

> **Gegeben:** Rechnende TM M_f, die eine berechenbare Funktion $f\colon \mathbb{N} \to \mathbb{N}$ berechnet.
> **Gesucht:** Der größtmögliche Ausgabewert von M_f.

(a) Wie lautet das zugehörige Entscheidungsproblem $\mathcal{X}_E$?

(b) Geben Sie eine Sprache über $\Sigma = \{0, 1\}$ an, deren Wortproblem $\mathcal{X}_E$ ist. Wie kodieren Sie hierbei die Eingabe?

(c) Wählen Sie aus der folgenden Liste die kleinste Komplexitätsklasse aus, die $\mathcal{X}_E$ enthält:

`P`, `NP`, `NPI`, `NPC`, `NP-schwer`, `PSPACE` $=$ {mit poly. Platzbedarf entscheidbare Probleme}, `DEC` $=$ {entscheidbare Probleme}, `UNDEC` $=$ {unentscheidbare Probleme}.

Aufgabe 9.2 Das Märchen kenne ich doch..?

Es war einmal eine Königstochter, die fuhr mit ihrem Kutscher hinaus in den Wald und setzte sich an einen Brunnen. Sie hatte eine goldene Kugel, die war ihr liebstes Spielzeug. Sie warf die Kugel ein ums andere Mal in die Luft und fing sie wieder auf. Doch da geschah es, dass sie die Kugel so hoch warf, dass sie ihr beim Fangen aus den Fingern glitt und in das Wasser fiel.
Die Königstochter blickte herab in den Brunnen, doch es war kein Grund zu sehen. Da steckte ein Frosch seinen Kopf aus dem Wasser und sprach: »Königstochter, was jammerst du so erbärmlich?« – »Ach, du garstiger Frosch kannst mir doch nicht helfen. Meine goldene Kugel ist mir in den Brunnen gefallen und das Wasser ist so tief, dass ich nicht an sie heran komme.«
Der Frosch sprach: »Deinen Reichtum verlange ich nicht, aber wenn du mich zum Gemahl nehmen willst, so will ich dir deine Kugel wiederbringen.« Die Königstochter dachte: »Ach, was will der einfältige Frosch von mir. Ein Frosch muss doch im Wasser bleiben. Aber er soll mir nur erst meine Kugel heraufholen.« Sie sprach: »Dir sei alles versprochen, was du verlangst, schaff mir nur erst meine goldene Kugel herbei.«
Sobald sie das gesagt hatte, tauchte der Frosch seinen Kopf wieder unter das Wasser, sank hinab und ein Weilchen später kam er wieder an die Oberfläche, hatte die goldene Kugel im Maul und warf sie heraus ins Gras. Da freute sich die Königstochter und lief beschwingt zurück zum Kutschwagen, der am Waldrand stehen geblieben war.

Als sie in die Kutsche stieg und sich noch einmal umdrehte, sah sie, dass der Frosch begonnen hatte, hinter ihr herzuhüpfen und dabei so geschwind wurde, dass sie ein Bangen ergriff. Die Königstochter rief ihrem Kutscher zu: »Gib Acht, dass wir nicht den direkten Weg zum Schloss meines Vaters einschlagen. Wir müssen um jeden Preis den Frosch abhängen, der mich verfolgt!« Nachdem sie einige Meter gefahren waren, fügt sie hinzu: »Ach guter Kutscher, am liebsten wäre es mir, wenn wir eine Strecke fahren, die so lange dauert wie möglich. Aber wir dürfen dabei keine

Stelle mehrmals passieren – nicht dass der Frosch sich zwischendurch niederlässt und uns beim wiederholten Vorbeifahren wieder auf den Fersen ist!« Der Kutscher griff zu seiner Straßenkarte, schaute auf die mit verschiedenen Weglängen bezeichneten Strecken und schlug verzweifelt die Hände über dem Kopf zusammen...

> FROSCHFLUCHT
> **Gegeben:** Ein ungerichteter Graph G mit Distanzen $d\colon E \to \mathbb{N}$ und Knoten $s, t \in V(G)$.
> **Gesucht:** Ein längster Pfad von s nach t, bei der kein Knoten mehrmals besucht wird.

Ist FROSCHFLUCHT unter Annahme $P \neq NP$ deterministisch in polynomieller Zeit lösbar?

Aufgabe 9.3 Eines haben wir noch!

Baba Yaga ist außer sich vor Wut. Nachdem sie von einem fünftägigen Ausflug in den Karpaten wieder in ihren wunderschönen Heimatwald zurückkehrt, sind beinahe alle Bäume abgestorben. Von den drei Buchen, die neben dem magische Brunnen standen ist bloß eine einzige übrig geblieben und selbst die massiven Eichen, die in einem engen Kreis um ihre Lieblingslichtung herum standen, sind plötzlich gestorben.
Die alte Hexe ruft ihr magisches Eichhörnchen zu sich: »Der Bäume müssen wieder her. Doch jetzt wo ich die Möglichkeit habe, werde ich den Wald so gestalten, wie ich es will – und Du hilfst mir dabei! Meine Magie wirkt dann am besten, wenn sich an verschiedenen Orten exakt die richtige Anzahl an Bäumen befindet. Ich gebe dir eine Liste mit meinen Wünschen und einen endlosen Vorrat an magischen Nüssen. Deine Aufgabe ist es nun, die Nüsse so einzupflanzen, dass all meine Wünsche erfüllt werden!«
Das magische Eichhörnchen schaut auf die Wunschliste der Baba Yaga. In der ersten Zeile steht: *An Brunnen, Lichtung, Quelle und Pilzkreis müssen insgesamt genau zwei Bäume gepflanzt werden.* Es schüttelt verwundert den Kopf und fragt nach: »Also kann ich dafür einfach zwei Nüsse neben dem Brunnen vergraben?« Da schreit Baba Yaga auf: »Natürlich nicht, du dumme Nuss! Auf keinen Fall dürfen an einem Ort mehrere magischen Nüsse gepflanzt werden! Vielleicht sollte ich doch besser den Troll von der Brücke damit beauftragen...«
Das will das magische Eichhörnchen nicht auf sich sitzen lassen und ruft: »Aber natürlich kannst du dich auf mich verlassen. Ich werde dir alle Wünsche auf dieser Liste erfüllen!«

> TREEPLANTINGSQUIRRELPROBLEM
> **Gegeben:** Eine Menge von Wünschen $\mathcal{W}$ und eine Menge von Orten $\mathcal{P}$. Ein Wunsch hat Form $w_i = (\mathcal{P}_i, k_i)$ mit $\mathcal{P}_i \subseteq \mathcal{P}$ und $k_i \in \mathbb{N}_+$ und ist erfüllt, wenn an *genau* k_i Orten aus $\mathcal{P}_i$ je eine Nuss platziert wurde.
> **Gefragt:** Gibt es eine Platzierung von Nüssen an Orten aus $\mathcal{P}$, sodass alle Wünsche in $\mathcal{W}$ gleichzeitig erfüllt sind?

Zeigen Sie, dass TREEPLANTINGSQUIRRELPROBLEM NP-vollständig ist.

Hinweis: Falls sie von 3-SAT kommen, überlegen Sie sich zuerst Variablen- und Klauselgadgets. Beide können sowohl aus neu erstellten Orten als auch Wünschen bestehen.

Aufgabe 9.4 Co-TSQP ist in NP?

Sie erinnern sich an das NP-vollständige TREEPLANTINGSQUIRRELPROBLEM (TSQP) vom letzten Übungsblatt. Wir betrachten das Problem Co-TSQP:

> **Gegeben:** Eine Menge von Wünschen $\mathcal{W}$ und eine Menge von Orten $\mathcal{P}$. Ein Wunsch hat Form $w_i = (\mathcal{P}_i, k_i)$ mit $\mathcal{P}_i \subseteq \mathcal{P}$ und $k_i \in \mathbb{N}_+$ und ist erfüllt, wenn an *genau* k_i Orten aus $\mathcal{P}_i$ je eine Nuss platziert wurde.
> **Gefragt:** Gilt für jede mögliche Platzierung von Nüssen an Orten aus $\mathcal{P}$, dass mindestens

ein Wunsch in $\mathcal{W}$ unerfüllt ist?

Der folgende Beweis, dass dieses Problem in *NP* liegt, ist falsch. Warum? Begründen Sie!

> *Beweis.* Sei `TSq` ein nicht-deterministischer polynomieller Entscheidungsalgorithmus für TSQP. Wir zeigen, dass es einen nicht-deterministischen polynomiellen Algorithmus `NonTSq` gibt, der Co-TSQP entscheidet:
>
> ```
> NonTSq(J):
>    bool res = TSq(J)
>    return !res
> ```
>
> Ist `res` = *false*, so war `TSq` nicht in der Lage eine Platzierung der Nüsse zu finden, also kann es keine solche geben. Schließlich hätte `TSq` ja richtig geraten, wenn es eine Platzierung gäbe. □

Aufgabe 9.5 Kernelization beim Elefantengeburtstag

Benjamin Blümchen feiert seinen n-ten Geburtstag und natürlich haben seine Freunde ihm eine große Geburtstagstorte mit n Kerzen mitgebracht. Genau wie bei den Menschen ist es auch im Tierreich Brauch, alle Kerzen auszupusten – jedoch nicht zwingenderweise alle auf einmal. Mit einem Luftstoß kann Benjamin beliebig viele Kerzen auspusten, vorausgesetzt sie stehen exakt hintereinander auf einer Geraden.

> KerzenAusblasen
> **Gegeben:** $n \in \mathbb{N}$ Kerzen mit Koordinaten in $\mathbb{N}^2$, $k \in \mathbb{N}$.
> **Gefragt:** Gibt es höchstens k Geraden, sodass jede der n Kerzen auf mindestens einer der Geraden liegt?

Geben Sie einen Algorithmus an, der in Polynomialzeit eine äquivalente Instanz erzeugt, deren Größe ausschließlich von k abhängt.

Aufgabe 9.6 Higher and higher

Gegeben ein randomisierter Polynomialzeitalgorithmus $\mathcal{A}$ für ein Entscheidungsproblem $\mathcal{X}$. Ist die Eingabe eine Nein-Instanz, gibt $\mathcal{A}$ immer korrekt `false` zurück; ist die Eingabe eine Ja-Instanz, gibt $\mathcal{A}$ mit Wahrscheinlichkeit p `true` zurück, sonst `false`.
Geben Sie jeweils einen randomisierten Polynomialzeitalgorithmus für $\mathcal{X}$ an, der Nein-Instanzen immer und Ja-Instanzen mit Wahrscheinlichkeit mindestens q korrekt entscheidet, wobei $q > p$.

(a) $p \geq \frac{1}{e}$,

(b) $p \geq \frac{1}{\mathrm{poly}(n)}$, wobei n die Kodierungslänge der Eingabe ist.

Hinweis zu Teil **(b)**: $\lim_{n\to\infty} \left(1 - \frac{1}{n}\right)^n = \frac{1}{e}$.