Mathematische Methoden der Algorithmik
Dr. Phillip Keldenich

Wintersemester 2026/2027
Abgabe: 17.11.2026
Besprechung: 24.11.2026

Übungsblatt 2

Abgabe der Lösungen bis zum 17.11.2026 um 13:15 Uhr in der großen Übung oder in den Hausaufgabenkasten der Algorithmik.

Gebt Eure Lösungen in Papierform über den Hausaufgabenabgabekasten ab (vor dem IZ 337). Denkt daran, Eure Zettel mit Namen und Matrikelnummer zu versehen. Elektronische Abgaben sendet Ihr bitte als *einzelnes* PDF-Dokument an `bossegra@ibr.cs.tu-bs.de`. Sollte die Abgabe mehrerer Dateien (z.B. Code) notwendig sein, versendet bitte ein einzelnes ZIP-Archiv. Bitte betrachtet die elektronische Abgabe als *Ausnahme*, z.B. im Krankheitsfall.

Aufgabe 1 (Simplex auf Dictionaries): **(15 + 15 Punkte)**
Löse die folgenden Probleme mit dem Zweiphasensimplex auf Dictionaries aus der Vorlesung. **Achtung:** Beachte bitte die angegebenen Regeln für die Wahl der *entering variable*. Diese ist je nach Teilaufgabe unterschiedlich; Abweichungen können zu deutlich mehr Pivotschritten mit hässlicheren Koeffizienten führen.

a) Löse das folgende LP mit dem Simplex auf Dictionaries aus der Vorlesung. Falls mehrere Variablen Kandidaten sind, die als *entering variable* gewählt werden könnten, wähle in jedem Schritt den Kandidaten x_j mit dem *niedrigsten Index* j. Schlupfvariablen haben dabei einen höheren Index (gemäß unserer Konvention $x_{n+i} \coloneqq w_i$).

$$\begin{aligned} \max \quad & 6x_1 + 8x_2 + 5x_3 + 9x_4 \\ \text{s.t.} \quad & 2x_1 + 1x_2 + 1x_3 + 3x_4 \leq 5 \\ & 1x_1 + 3x_2 + 1x_3 + 2x_4 \leq 3 \\ & x_1, x_2, x_3, x_4 \geq 0. \end{aligned}$$

b) Löse das folgende LP mit dem Zweiphasensimplex aus der Vorlesung. Falls mehrere Variablen Kandidaten sind, die als *entering variable* gewählt werden können, wähle in jedem Schritt den Kandidaten x_j mit dem *höchsten Koeffizienten* $\bar{c}_j$ in der aktuellen Formulierung der Zielfunktion. Auf den ersten Pivotschritt für Phase I hat diese Regel keinen Einfluss; hier ist stets x_0 die *entering variable* und die Variable mit dem niedrigsten $\bar{b}_i$ die *leaving variable*.

$$\begin{aligned} \max \quad & 2x_1 - 6x_2 \\ \text{s.t.} \quad & -x_1 - x_2 - x_3 \leq -2 \\ & 2x_1 - x_2 + x_3 \leq 1 \\ & x_1, x_2, x_3 \geq 0. \end{aligned}$$

Aufgabe 2 (Laufzeit von Simplex): **(5 + 5 + 10 + 10 Punkte)**

Ein wichtiger Aspekt des Simplex-Algorithmus wurde in der Vorlesung noch nicht besprochen: die Laufzeit. In dieser Aufgabe wollen wir die Laufzeit einer einzelnen Iteration auf einem Dictionary zu einem LP in Standardform mit n ursprünglichen Variablen und m Nebenbedingungen (≥ 0-Nebenbedingungen nicht eingerechnet) betrachten.

Wir nehmen dabei an, dass wir bereits ein gültiges Dictionary haben; entweder ist also Phase I schon gelaufen oder die triviale Startbasis führt zu einer zulässigen Startlösung. Weiter nehmen wir folgendes an: anders als in der Praxis achten wir zunächst nicht besonders auf Koeffizienten mit Wert 0. Unser Dictionary ist gespeichert als ein Array von m Arrays mit je n Elementen plus einen Vektor $\bar{c}$ mit n Elementen sowie einen Vektor $\bar{b}$ mit m Elementen, und soll nach dem Pivotschritt auch wieder diese Form haben. Wir haben darüber hinaus eine Datenstruktur (wie z.B. eine Hashtabelle oder entsprechende Arrays), die es uns erlaubt, in Zeit $O(1)$ den Zeilenindex zu finden, der zu einer Basisvariable gehört, bzw. den Spaltenindex zu finden, der zu einer Nichtbasisvariable gehört sowie andersherum die Basis- und Nichtbasisvariable zu einem gegebenen Index, und wir können diese Datenstruktur in konstanter Zeit updaten, um eine Basisvariable und Nichtbasisvariable zu tauschen.

a) Welche Laufzeit haben die beiden Regeln zur Wahl der *entering variable* aus Aufgabe 1 (niedrigster Index bzw. größter Koeffizient $\bar{c}_j$)?

b) Gegeben eine Wahl x_j für die *entering variable*, welche Laufzeit hat die Bestimmung des neuen Werts von x_j bzw. die Erkennung der Unbeschränktheit sowie das Finden einer beliebigen (z.B. der ersten) *leaving variable*?

c) Mal angenommen, wir wollten stattdessen diejenige *entering variable* wählen, welche den Wert der Zielfunktion nach der Pivotoperation maximiert. Welche Laufzeit hätte eine naive Implementierung dieser Auswahlstrategie?

d) Nach der Wahl von *entering variable* x_j und *leaving variable* x_k müssen wir die alte Definition von x_k nach x_j umstellen und überall für x_j die neue Definition einsetzen. Beschreibe, wie diese Operation mit nur $O(n)$ zusätzlichem Speicherplatz auf unseren Arrays implementiert werden kann. Welche Laufzeit haben diese Operationen?