Marburg University

Department of Computer Science
Prof. Dr. Sebastian Wild

Date: 2025-01-23
Version: 2026-01-23 15:35

Übungsblatt 12 zur Vorlesung Effiziente Algorithmen (Winter 2025/26)

Abgabe: *Bis 2026-01-30 18:00, on ILIAS.*

1. Aufgabe

20 + 20 + 10 Punkte

a) Zeigen oder widerlegen Sie: Sei $G = (V, E)$ ein Graph mit Kantengewichten $\ell_e \geq 0$. Weiterhin sei P ein kürzester Weg von s nach t, also eine Lösung des kürzeste Wege Problems. Dann bleibt P eine Lösung des Problems, wenn wir alle Kantenlängen ℓ_e durch ihre Quadrate ${\ell_e}^2$ ersetzen.

b) Sei $G = (V, E)$ ein gerichteter Graph, der keine negativen Zyklen enthält. Zeigen Sie: Ist $(s = v_0, v_1, \ldots, v_k = t)$ ein kürzester *Knoten-einfacher* Weg (kürzester Pfad) von s nach t, so ist auch jeder Teilweg $(v_0, \ldots, v_l)$ für $l = 1, \ldots, k-1$ ein kürzester *Knoten-einfacher* Weg (Pfad) von s nach v_l.

c) Zeigen Sie, dass diese Aussage b) im Allgemeinen nicht gilt, wenn der Graph negative Zyklen enthält.

2. Aufgabe

30 Punkte

Ein Pfad-Graph ist ein ungerichteter Graph $P = (V, E)$, wobei

$$V = \{v_1, \ldots, v_n\}$$
$$E = \{\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}\}$$

Betrachten Sie folgendes Problem:

Gegeben sei ein Pfad-Graph $P = (V, E)$ und eine Kantenfunktion $c : E \to \{\text{rot}, \text{blau}\}$, welche einer Kante eine Farbe zuweist. Bestimmen Sie die minimale Anzahl von Kanten, die aus P gelöscht werden müssen, damit kein Knoten mit einer roten und einer blauen Kante inzident ist.

Entwickeln Sie einen Greedy-Algorithmus, der das Problem in $\mathcal{O}(n)$ Zeit löst. Zeigen Sie, dass Ihr Algorithmus korrekt ist.

(Hinweis: In der Präsenzübung wird eine mögliche Greedy-Strategie widerlegt.)

3. Aufgabe

40 Punkte

Sei S eine endliche Menge mit $|S| \geq 2$. Seien weiterhin mit $\{S_1, \ldots, S_k\}$ eine Partition von S und mit $\mathcal{U} = \{A \subseteq S \mid |A \cap S_i| \leq 1, 1 \leq i \leq k\}$ eine Menge von Teilmengen gegeben.

Zeigen Sie: $(S,\mathcal{U})$ ist ein Matroid. Beachten Sie dabei, das ein *hereditary set system* nicht-leer sein muss.

4. Aufgabe

20 + 10 + 30 Punkte

Betrachten Sie folgendes Problem:

Gegeben eine Zeichenkette s und eine Liste von Wörtern w, überprüfen Sie, ob s durch Einfügen von Leerzeichen in eine Sequenz von Wörtern aus w umgewandelt werden kann.

Beispiel:
$s =$ `platzangstübersehen`
$w = [$`angst`, `über`, `unter`, `weit`, `sehen`, `platz`, `platzangst`, `wir`, `übersehen`$]$

Mögliche Lösungen:

- `platz angst über sehen`
- `platzangst übersehen`

Folgender Algorithmus löst das Problem rekursiv:

```
1 procedure WordRek(s, w):
2     if s.isEmpty()
3         return True
4     for i := 1, ..., |s|
5         pre := s.substring(0, i)
6         suf := s.substring(i + 1, |s|)
7         if w.contains(pre) ∧ WordRek(suf, w)
8             return True
9     return False
```

a) Nehmen Sie vereinfachend an, dass der Aufruf von *contains* und *isEmpty* immer die Kosten c hat, und, dass das Bilden von Pre- und Suffixen keine weiteren Kosten erzeugt. Wie können Sie die Komplexität der obigen Lösung abschätzen?

b) Der vorgestellte Algorithmus enthält Teilprobleme, die immer wieder berechnet werden. Diskutieren Sie, welche Teilprobleme das sind.

c) Entwickeln Sie einen Algorithmus auf Basis von dynamischen Programmieren, welcher das Problem in $\mathcal{O}(n^2)$ Zeit und $\mathcal{O}(n)$ Speicherbedarf löst. Wenden Sie hierbei den **Bottom-Up**-Ansatz an.