

Allgemeine Hinweise

- Zusätzliche freiwillige Aufgaben werden diese Woche als **separates Weihnachtsblatt** erscheinen. Bitte sehen Sie morgen oder übermorgen noch einmal auf der Homepage der Veranstaltung nach. Sie können durch eine Bearbeitung der freiwilligen Aufgaben zusätzliche Punkte bekommen.

9.1 Graphen färben (5 Punkte)

Sei $G = (V, E)$ ein ungerichteter Graph. Wir nennen G k -färbbar, wenn man seine Knoten so mit k Farben markieren kann, dass keine zwei benachbarten Knoten die gleiche Farbe haben.

- (3P) Beschreiben Sie einen Algorithmus, der mittels Breitensuche feststellt, ob ein Graph 2-färbbar ist.
- (2P) Geben Sie Pseudocode für Ihre Beschreibung an. Als Ausgangsbasis dient dabei der Code aus der Vorlesung.

9.2 Wortwege (15 Punkte)

Gegeben sei eine Liste aller englischen Wörter mit vier Buchstaben. Diese bilden als Knoten zusammen mit folgenden Kanten einen Graph: Mit einer Kante verbunden seien alle Wörter, bei denen sich an exakt einer Stelle ein Buchstabe unterscheidet. Da die Kanten der Aufgabe ungerichtet sind, repräsentieren Sie diese jeweils durch eine Hin- und eine Rückkante. Beantwortet werden soll die Frage, ob es einen Weg von einem Start-Wort s zu einem Ziel-Wort z gibt.

Beispiel: $s = \text{abys} \rightarrow \text{abye} \rightarrow \text{able} \rightarrow \text{axle} = z$

- (1P) Weshalb eignet sich für diesen Graph eine Adjazenz-Liste besser als eine -Matrix?
- (5P) Laden Sie sich die Datei `4LetterWords.txt` herunter und schreiben Sie ein Java-Programm, dass die Datei lädt und den Graph als Adjazenzliste im Speicher erstellt.
- (5P) Implementieren Sie eine Breitensuche, wie in der Vorlesung besprochen. Finden Sie damit den obigen Beispielweg und geben Sie diesen aus. Schreiben Sie dazu nacheinander alle zum Weg gehörenden Knoten, inkl. Anfang und Endknoten, zeilenweise auf die Konsole.
- (4P) Finden Sie den Längsten aller kürzesten Wege im Graph und geben Sie diesen aus. Falls der längste Weg nicht eindeutig ist, geben Sie einen der längsten Wege aus.