

Station 3 – Imperative Programmierung

Entwerfen Sie Algorithmen für die folgenden zwei Probleme. Die geforderten Algorithmen sollen in Pseudocode aufgeschrieben werden. Dieser muss dementsprechend verfeinert werden, dass eine Übertragung in eine imperative Programmiersprache ohne weitere kreative Schritte möglich ist.

1. **Name:** UnidirektionaleKantenZählen

Eingabe: Gerichteter Graph $G = (V, E)$ in Adjazenzlisten-Repräsentation

Ausgabe: Anzahl unidirektionaler Kanten¹ in G

zähler $\leftarrow 0$

für alle $v_1 \in V$

| für alle $v_2 \in \text{adj}[v_1]$

| | falls $v_1 \notin \text{adj}[v_2]$

| | | zähler \leftarrow zähler + 1

gib zähler aus

2. **Name:** ListenVergleich

Eingabe: Zwei Listen L_1 und L_2 , deren Elemente positive ganze Zahlen sind

Ausgabe: 1, falls L_2 ein Element enthält, das in L_1 nicht enthalten ist und 0 sonst

für alle $i \in L_2$

| falls $i \notin L_1$

| | gib 1 aus

| | STOP

gib 0 aus

¹ Eine Kante $(e_1, e_2) \in E$ heißt unidirektional, wenn $(e_2, e_1) \notin E$.