11. Aufgabenblatt zur Vorlesung

Algorithmen und Datenstrukturen SoSe 2025

Wolfgang Mulzer

Abgabe am 11. Juli 2025 bis 10 Uhr im Whiteboard
Dies ist das letzte Aufgabenblatt.

Aufgabe 1 Tiefensuche *10 Punkte*

(a) Geben Sie Pseudocode für einen Algorithmus, der das folgende Problem löst: Gegeben ein zusammenhängender ungerichteter Graphen G, finde eine Weg, der jede Kante in G genau einmal in jeder Richtung durchläuft.

Hinweis: Verwenden Sie Tiefensuche.

(b) Sei $G = (V, E)$ ein Graph. Gegeben ein Algorithmus, der in $O(|V|)$ Zeit überprüft, ob G *genau einen* Kreis enthält.

Hinweis: Der Graph G muss nicht zusammenhängend sein. Wie müssen die Komponenten von G aussehen, wenn G genau einen Kreis enthält? Was haben Sie in Diskrete Strukturen über Bäume gelernt?

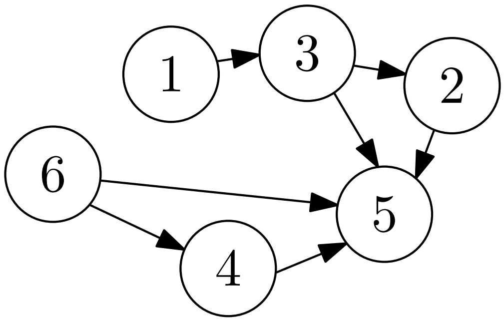
Aufgabe 2 Topologisches Sortieren *10 Punkte*

Sei $G = (V, E)$ ein kreisfreier gerichteter Graph. Ein solcher Graph heißt DAG (für **d**irected **a**cyclic **g**raph). Eine *topologische Sortierung* von G ist eine Anordnung der Knoten von G, so dass keine Kante in G von einem späteren zu einem früheren Knoten verläuft. Betrachten Sie den folgenden Algorithmus:

```
dfs(v, topoOrder)
    v.found <- true
    for e in v.outgingEdges() do
        w <- e.opposite(v)
        if !w.found then
            dfs(w, topoOrder)
    topoOrder.push(v)

topoSort(G)
    topoOrder <- new Stack
    for v in G.vertices() do
        v.found <- false
    for v in G.vertices() do
        if !v.found do
            dfs(v, topoOrder)
```

(a) Führen Sie den Algorithmus auf dem untigen Graphen aus. Gehen Sie dabei davon aus, dass die Knoten in `vertices` und `outgoingEdges` gemäß der Knotennummern angeordnet sind. Zeigen Sie die einzelnen Schritte. Wie erhält man im Anschluss die topologische Sortierung?



(b) Beweisen Sie, dass der Algorithmus aus (a) funktioniert.

Hinweis: Führen Sie einen Widerspruchsbeweis. Nehmen Sie an, es gäbe eine Kante e von einem späteren zu einem früheren Knoten in `topoSort`, und überlegen Sie sich, was passiert, wenn `dfs` die Kante e untersucht. Beachten Sie dabei, dass es einen Unterschied macht, ob die Tiefensuche für den Endpunkt von e noch aktiv ist.

Aufgabe 3 Kürzeste Wege in DAGs *10 Punkte*

Sei $G = (V, E)$ ein gewichteter gerichteter azyklischer Graph. Seien $s, t \in V$. Geben Sie einen Algorithmus, der einen kürzesten Weg von s nach t in Zeit $O(|V| + |E|)$ berechnet. Beweisen Sie die Korrektheit und die Laufzeit Ihres Algorithmus.

Hinweis: Beachten Sie Aufgabe 2(b) und verfahren Sie ähnlich zum Algorithmus von Dijkstra.