Algorithmen & Datenstrukturen 1

Sommersemester 2022

Prof. Dr. Martin Hoefer
Tim Koglin, Marco Schmalhofer

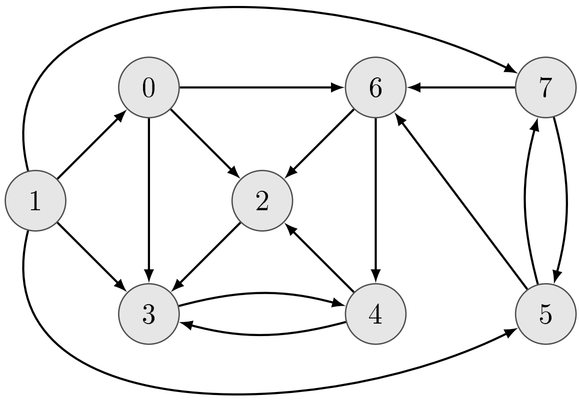
GOETHE UNIVERSITÄT FRANKFURT AM MAIN
Institut für Informatik
Algorithmen und Komplexität

Übungsblatt 8

Ausgabe: 07.06.2022
Abgabe: 14.06.2022, **08:00**

Aufgabe 8.1 *Tiefen- und Breitensuche* (7 + 7 Punkte)

Gegeben sei folgender Graph G.



a) Führen Sie eine Tiefensuche auf G aus. Besuchen Sie dabei die Nachbarn der Knoten in aufsteigender Reihenfolge.

   Geben Sie den Wald der Tiefensuche an und klassifizieren Sie alle Kanten von G als Baum-, Rückwarts-, Vorwärts- oder Querkanten.

b) Führen Sie eine Breitensuche auf G beginnend in Knoten 0 aus. Fügen Sie die Nachbarn der Knoten jeweils in aufsteigender Reihenfolge in die Queue ein.

   Geben Sie den Baum der Breitensuche an. Klassifizieren Sie für die Breitensuche alle besuchten Kanten von G analog zur Definition der Kantentypen der Tiefensuche.

Aufgabe 8.2 *Adas Dienstreise* (9 Punkte)

Ada ist auf Dienstreise in der idyllischen Stadt Puerto Algorico. Da sie in der Arbeitsstätte vor Ort sehr viel erledigen muss, hat sie praktisch keine Zeit, etwas von der besonderen Architektur der Stadt zu sehen. Daher möchte Ada herausfinden, ob sie zumindest den Weg von ihrem Hotel zur Arbeitsstätte etwas abwechslungsreicher gestalten kann, ohne dass der Weg mehr Zeit in Anspruch nimmt als nötig.

Helfen Sie Ada, indem Sie einen Algorithmus in Pseudocode entwerfen, der für ein gegebenes gerichtetes Straßennetzwerk $G = (V, E)$ bestimmt, wie viele verschiedene kürzeste Wege es von Adas Hotel zur Arbeitsstätte gibt. Dabei gelten zwei Wege als verschieden, wenn sie nicht identisch sind. Ada weiß, dass sie für alle Straßen in E die gleiche Zeit benötigt, um sie zu durchgehen. Ihr Algorithmus sollte die Laufzeit $\mathcal{O}(|V| + |E|)$ nicht überschreiten. Beschreiben Sie wie immer zuerst Ihre Idee und zeigen Sie, dass Ihr Algorithmus korrekt funktioniert sowie die geforderte Laufzeitschranke einhält.

Aufgabe 8.3 *Alans Dienstreise* (9 Punkte)

Auch Alan befindet sich auf Dienstreise, im hügeligen Nachbarort Las Datas. Ihm ist im Gegensatz zu Ada nicht wichtig, kürzeste Wege zu nehmen, sondern dass er nach einem arbeitsintensiven Tag in seiner Arbeitsstätte einen *erholsamen* Weg von dort mit dem Fahrrad zurück zu seinem Hotel hat. Das bedeutet, dass er stets strikt bergab rollen und niemals in die Pedale treten will. Alan möchte dabei verschiedene solcher erholsamen Wege ausprobieren.

Helfen Sie Alan, indem Sie einen Algorithmus in Pseudocode entwerfen, der für ein gegebenes gerichtetes Straßennetzwerk $G = (V, E)$ bestimmt, wie viele verschiedene erholsame Wege es zwischen Alans Arbeitsstätte und seinem Hotel gibt. Dabei gelten wieder zwei Wege als verschieden, wenn sie nicht identisch sind. Ihr Algorithmus sollte die Laufzeit $\mathcal{O}(|V|+|E|)$ nicht überschreiten. Beschreiben Sie wie immer zuerst Ihre Idee und zeigen Sie, dass Ihr Algorithmus korrekt funktioniert sowie die geforderte Laufzeitschranke einhält.

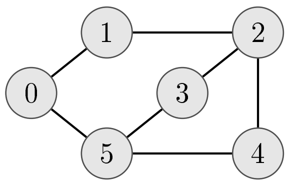
Hinweis: Im Kontext dieser Aufgabe bedeutet eine gerichtete Kante $(u, v) \in E$, dass es von Knoten u zu Knoten v strikt bergab geht.

Aufgabe 8.4 *Begrünung* (2 + 6 Punkte)

Um die Stadt zu begrünen, beschließt die Verwaltung eines Frankfurter Stadtteils, großflächig Grünstreifen anzulegen. Zu diesem Zweck sollen die bisher ausnahmslos in beide Fahrtrichtungen nutzbaren Straßen komplett in ein System von Einbahnstraßen umgewandelt werden. Danach soll es aber weiterhin möglich sein, von jedem Ort in der Stadt zu jedem anderen zu gelangen.

Um obiges Problem zu lösen, modellieren wir das Straßensystem als zunächst ungerichteten Graphen $G = (V, E)$. Jeder Kante soll nun eine Richtung zugewiesen werden, so dass der resultierende gerichtete Graph $\vec{G} = (V, \vec{E})$ stark zusammenhängend ist.

a) Betrachten Sie folgenden ungerichteten Graphen G:



Weisen Sie jeder Kante in G eine Richtung zu, so dass der resultierende gerichtete Graph $\vec{G}$ stark zusammenhängend ist.

b) Sei G nun ein beliebiger ungerichteter Graph. Es gilt die folgende Aussage:

> Ein stark zusammenhängendes Einbahnstraßennetz $\vec{G}$ existiert genau dann, wenn der Ausgangsgraph G zusammenhängend ist und keine *Brücke* existiert.

Eine *Brücke* ist eine Kante, durch deren Löschung der Graph in zwei Zusammenhangskomponenten zerfällt. Sie dürfen diese Aussage im Folgenden ohne Beweis verwenden.

Nehmen Sie an, G sei zusammenhängend und habe keine Brücken. Entwerfen Sie einen Algorithmus, der jeder Kante in G eine Richtung zuweist, so dass der resultierende Graph $\vec{G}$ stark zusammenhängend ist. Ihr Algorithmus sollte die Laufzeit $\mathcal{O}(|V| + |E|)$ nicht überschreiten.

Beschreiben Sie zuerst Ihre Idee und dann den Algorithmus (Pseudocode ist nicht erforderlich). Zeigen Sie die Korrektheit Ihres Algorithmus und begründen Sie, warum die Laufzeitschranke eingehalten wird.

Bei allgemeinen Anmerkungen zu den Übungsaufgaben oder Fragen zum Übungsbetrieb erreichen Sie uns unter der folgenden E-Mail-Adresse: algo122@cs.uni-frankfurt.de.