Übungsblatt 10

Algorithmen I – Sommersemester 2023

Abgabe im ILIAS bis 07.07.2023, 18:00 Uhr
Die Abgabe erfolgt als *eine* PDF-Datei über das Übungsmodul in der Gruppe deines Tutoriums im ILIAS.
Beachte bitte die Hinweise zum Bearbeiten auf der Webseite.

Aufgabe 1 - Lauter Bäume (0+3 Punkte)

Hinweis: Diese Aufgabe bringt ausschließlich Bonuspunkte.
Entscheide für jede Aussage, ob sie wahr oder falsch ist und begründe deine Antwort.

a*) Wir nennen einen gerichteten Graphen $G = (V, E)$ *semi-zusammenhängend*, wenn für alle Paare von Knoten $u, v \in V$ gilt, dass u von v aus erreichbar ist oder v von u aus (oder beides). Jeder gerichtete, azyklische Graph mit eindeutiger Quelle ist semi-zusammenhängend. (1 Punkt)
Hinweis: Eine Quelle ist ein Knoten ohne eingehende Kanten.

b*) Sei $G = (V, E)$ ein ungerichteter Graph und $\{v, w\} \in E$ eine Kante von G. Dann ist v in jedem DFS-Baum von G Nachfahre oder Vorfahre von w. (1 Punkt)

c*) Sei G ein Graph und v ein Knoten in G. Dann ist jeder DFS-Baum mit Wurzel v mindestens so hoch wie der BFS-Baum mit Wurzel v. (1 Punkt)

Aufgabe 2 - Cut-Vertices (9 Punkte)

In einem Graph $G = (V, E)$ bezeichnen wir einen Knoten $v \in V$ als *Cut-Vertex*, wenn G durch Löschen von v in mehrere Zusammenhangskomponenten zerfällt. In dieser Aufgabe wollen wir alle Cut-Vertices eines Graphen mithilfe von DFS finden.

a) Zeige, dass die Wurzel s eines DFS-Baums genau dann ein Cut-Vertex von G ist, wenn sie mehr als ein Kind hat. (3 Punkte)

b) Zeige, dass ein Knoten v, der nicht die Wurzel des DFS-Baums ist, genau dann ein Cut-Vertex von G ist, wenn gilt: v hat ein Kind u im DFS-Baum, sodass es keine Rückkante aus dem Teilbaum unter u zu Vorfahren von v gibt. (2 Punkte)

c) Beschreibe, wie mithilfe der low-Werte entschieden werden kann, ob ein (nicht Wurzel-) Knoten ein Cut-Vertex ist. Begründe deine Antwort. (1 Punkt)

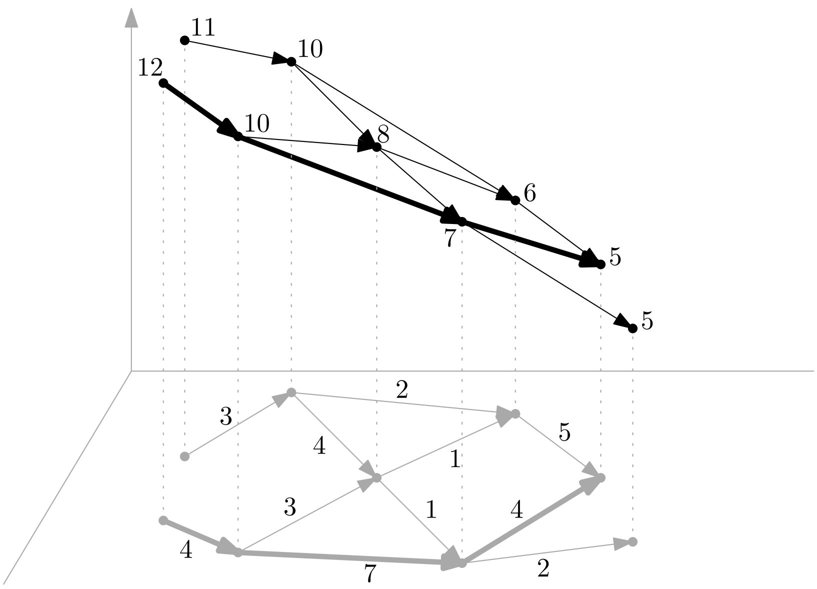
d) Gib einen Algorithmus an, der in $O(n+m)$ Zeit alle Cut-Vertices in G bestimmt. Begründe dessen Korrektheit und dass er die geforderte Laufzeit hat. (3 Punkte)

Aufgabe 3 - Freibad (7 Punkte)

Hinweis: Diese Aufgabe baut teilweise auf Inhalten von Vorlesung 15 auf.
Um sich von den Strapazen ihrer "Arbeit" zu erholen, nehmen sich die Biber eine Auszeit im Freibad. Dort gibt es eine aufregende Wasserrutsche mit mehreren Ab*zweig*ungen und Knotenpunkten. Die Abschnitte der Rutsche haben unterschiedliche Spaßfaktoren und je nachdem wie man abbiegt, macht die Rutsche insgesamt mehr oder weniger Spaß.

Wir modellieren die Wasserrutsche als gerichteten, gewichteten Graphen $G = (V, E)$. Jeder der n Knoten hat eine geographische Höhe $h : V \to \mathbb{N}$. Jede der m Kanten $(u, v) \in E$ repräsentiert einen Abschnitt der Rutsche, wobei klar ist, dass $h(u) > h(v)$ gilt, da Wasser nur bergab fließen kann. Zudem ist das Gewicht einer Kante e durch den *Spaßfaktor* $\text{fun}(e) \in \mathbb{N}$ gegeben. Im folgenden bezeichnen wir einen Pfad, der das Rutschnetzwerk herab geht als *Talweg* und der *Spaß* eines Talweges ist gegeben durch die Summe der Spaßfaktoren der enthaltenen Kanten.

Die Biber sind bei der Wahl ihrer Abzweigungen ziemlich gierig. An jedem Knotenpunkt wählen sie als nächsten Abschnitt einfach den, dessen Spaßfaktor maximal ist.



Ein Rutschnetzwerk mit Knotenhöhen in schwarz. In grau das gleiche Netzwerk von oben betrachtet mit den zugehörigen Spaßfaktoren. Ein gieriger Talweg mit Spaß 15 ist fett markiert.

Die Biber glauben damit glücklich zu sein, aber scheinbar willst du ihnen das kaputt machen. Dir fällt auf, dass der Spaß eines *gierigen Talweges* viel kleiner sein kann, als der *absolut spaßigste Talweg (AST)*, welcher die Summe der Spaßfaktoren maximiert.

a) Reibe es den Bibern unter die Nase, indem du für jedes $k \in \mathbb{N}$ einen Graphen angibst, bei dem der AST mindestens k-mal spaßiger ist als ein gieriger Talweg. (1 Punkt)

Das Resultat: traurige Biber. Dr. Meta ist sauer und verlangt, dass du einen Algorithmus entwickelst, der für ein gegebenes Rutschnetzwerk den AST findet. Wie üblich, ist es hilfreich, das Problem auf ein kleineres zu reduzieren, was sich leichter lösen lässt.

b) Sei $v \in V$ ein Knoten. Angenommen, wir kennen für jeden Nachbarknoten $w \in N(v)$ bereits den Spaß $\text{astfun}(w)$ des ASTs, der bei w beginnt. Gib an, wie wir damit den Spaß $\text{astfun}(v)$ des ASTs berechnen können, der bei v beginnt. (1 Punkt)

Mithilfe dieser Lösung für Teilprobleme wollen wir nun einen Algorithmus entwickeln der das Problem auf dem gesamten Graphen löst. Die Frage ist, in welcher Reihenfolge wir die Knoten betrachten sollten ...

c) Zeige, dass das Rutschnetzwerk ein gerichteter, *azyklischer* Graph ist. (1 Punkt)

d) Welche Reihenfolge der Knoten bietet sich nun an, um wie in Teilaufgabe b) den AST eines Knotens aus den ASTs seiner Nachbarn herzuleiten? (1 Punkt)

e) Angenommen, die Knoten liegen in der Reihenfolge aus Teilaufgabe d) vor. Beschreibe einen Algorithmus, der in $O(n+m)$ Zeit den AST von G findet. Begründe, wieso der Algorithmus korrekt ist und die geforderte Laufzeit hat. (3 Punkte)

Aufgabe 4 - ProofGPT (3 Punkte)

Hinweis: Diese Aufgabe baut teilweise auf Inhalten von Vorlesung 15 auf.
Musterlösungen für Übungsaufgaben zu schreiben ist anstrengend. Deswegen lagern wir das jetzt an generative, künstliche Intelligenz aus. Die meisten Lösungen zu diesem Blatt sind so entstanden. Wird schon passen. Ganz uneigennützig gestatten wir euch in dieser Aufgabe einen Blick hinter die Kulissen. Wir beschäftigen uns mit folgender Aussage:

> *Sei $G = (V, E)$ ein gerichteter Graph und seien $u, v \in V$. Angenommen, es gibt einen Pfad von u nach v in G. Dann liegt v in jedem an $s \notin \{u, v\}$ gewurzelten DFS-Baum von G im Teilbaum unter u.*

Auch wenn wir zwei verschiedene Beweise generieren konnten (siehe unten), sind wir uns doch unsicher, ob wir diesen trauen können. Leider ist das Kontrollieren von KI-generierten Lösungen ziemlich anstrengend, weswegen wir das jetzt an dich auslagern.

a) Benenne für die beiden “Beweise” jeweils eine Stelle, an der sie kaputt gehen und begründe deine Antworten. (2 Punkte)

b) Entscheide, ob die Aussage wahr oder falsch ist und begründe deine Antwort. (1 Punkt)

Beweis durch Widerspruch:

1. Nehmen wir an, es existiert ein DFS-Baum mit der Wurzel s, in dem v nicht in einem Teilbaum unterhalb von u liegt.
2. Betrachten wir die Entdeckungs- und Abschlusszeiten von u, v und s im DFS-Durchlauf.
3. Da es einen Pfad von u nach v in G gibt, gibt es drei Möglichkeiten für das Verhältnis zwischen den Entdeckungs- und Abschlusszeiten von u, v und s:
   a. Fall 1: v wird vor u entdeckt und abgeschlossen: In diesem Fall würde v im Teilbaum unterhalb von u liegen, was unserer Annahme widerspricht.
   b. Fall 2: u wird vor v entdeckt und abgeschlossen: Dieser Fall ist nicht möglich, da v in dem angenommenen DFS-Baum mit Wurzel s nicht im Teilbaum unterhalb von u liegt.
   c. Fall 3: v wird vor u entdeckt, aber nach u abgeschlossen: Auch dieser Fall ist nicht möglich, da v in einem gültigen DFS-Baum vor u abgeschlossen sein sollte.
4. Aus der Analyse der Möglichkeiten folgern wir, dass es keinen DFS-Baum mit Wurzel s geben kann, in dem v nicht in einem Teilbaum unterhalb von u liegt.
5. Daher gilt die Aussage, dass in jedem DFS-Baum mit Wurzel s v in einem Teilbaum unterhalb von u liegen muss.

Durch Widerspruch haben wir gezeigt, dass die Annahme, dass v nicht in einem Teilbaum unterhalb von u in einem DFS-Baum mit Wurzel s liegt, zu einem Widerspruch mit den Regeln der Tiefensuche führt. Somit ist die Aussage bewiesen.

Dieses Ergebnis bestätigt, dass in jedem gültigen DFS-Baum, wenn es einen Pfad von u nach v gibt, v immer in einem Teilbaum unterhalb von u liegt, unabhängig von der Wahl des Wurzelknotens s.

"Beweis" 1: Die "Entdeckungszeit" wird in der Vorlesung als DFS-Nummer bezeichnet. Die "Abschlusszeit" ist die FIN-Nummer.

Selbstverständlich! Hier ist ein alternativer Beweis für die gegebene Aussage:

Wir nehmen an, es gibt einen Pfad von u nach v in einem gerichteten Graphen G, und wir betrachten einen beliebigen DFS-Baum von G mit s als Wurzel. Wir möchten zeigen, dass v im Teilbaum unterhalb von u liegt.

Wir führen den Beweis per Induktion über die Länge des Pfades von u nach v.

Induktionsbasis:
Wenn der Pfad von u nach v eine Länge von 1 hat, bedeutet das, dass es eine direkte Kante von u nach v gibt. In diesem Fall ist v ein direkter Nachfolger von u im DFS-Baum und liegt daher im Teilbaum unterhalb von u.

Induktionsschritt:
Wir nehmen an, die Aussage gilt für einen Pfad der Länge k. Nun betrachten wir einen Pfad von u nach v mit einer Länge von k+1. Wir können diesen Pfad in zwei Teile aufteilen: den Pfad von u nach w mit einer Länge von k und die Kante von w nach v.

Da der Pfad von u nach w eine Länge von k hat, nach der Induktionsannahme liegt w im Teilbaum unterhalb von u im DFS-Baum. Da v der Nachfolger von w ist, liegt v im Teilbaum unterhalb von w.

Da v im Teilbaum unterhalb von w liegt und w im Teilbaum unterhalb von u liegt, folgt daraus, dass v im Teilbaum unterhalb von u liegt.

Somit wurde gezeigt, dass v im Teilbaum unterhalb von u im DFS-Baum liegt, wenn es einen Pfad von u nach v in G gibt.

“Beweis” 2: Ich weiß ja nicht …

Eulenfest am 13. Juli 2023

In zwei Wochen am Donnerstag findet endlich wieder das Eulenfest statt! Los geht es um 18:30 Uhr, im und um den Infobau. Dieses Jahr gibt es Livemusik von den Bands *Jon Doe* und *Carry on Atlas*. Natürlich haben wir auch wieder DJs und Karaoke. Außerdem werdet ihr mit Getränken und Essen versorgt. Weitere Infos zum Fest findet ihr unter eulen.party.

Gerne könnt ihr uns mit einer Helferschicht unterstützen. Tragt euch dafür einfach auf der Website ins Helfersystem ein.

Wir suchen auch noch tatkräftige Unterstützung für unser Sicherheitsteam. Hierfür brauchst du keine besonderen Qualifikationen oder viel Erfahrung, es wird aber vor dem Fest eine ausführliche Einweisung geben. Falls du Interesse hast, melde dich bei Max (max.vogel@fsmi.uni-karlsruhe.de) per Mail. Ebenfalls suchen wir Personen mit Sanitätsausbildung, hierfür kannst du dich gerne bei Patrick (patrick.schneider@fsmi.uni-karlsruhe.de) melden.

Mehr Infos