

Übungsblatt 11

Algorithmen I – Sommersemester 2022

Abgabe im ILIAS bis 13.07.2022, 14:00 Uhr

Bitte beschrifte Deine Abgabe gut sichtbar mit Deinem Namen und Deiner Matrikelnummer. Achte insbesondere bei handschriftlichen Abgaben auf Lesbarkeit und genügend Platz für Korrektur-Anmerkungen. Die Abgabe erfolgt über das Übungsmodul in der Gruppe Deines Tutoriums im ILIAS. Gib Deine Ausarbeitungen in *einer* PDF-Datei ab. Achte darauf, effiziente Algorithmen zu formulieren, also solche mit möglichst geringer asymptotischer Laufzeit!

Wenn du die Korrektheit eines Algorithmus begründen oder dessen Laufzeit analysieren sollst, tue dies getrennt von der Beschreibung des Algorithmus.

Wenn nicht anders spezifiziert oder aus dem Kontext ersichtlich, bezeichnen wir mit Graph einen einfachen ungerichteten Graphen.

Aufgabe 1 - Kann ja mal passieren... (10 Punkte)

Während seiner Undercover-Aktion auf der Mitarbeiterversammlung ist Dr. Meta aufgefallen, dass er bei der Ausdünnung ein relativ wichtiges Detail übersehen hatte: Beim Modellieren des Kontakt-Graphen war ihm entgangen, dass dieser eigentlich gerichtet hätte sein müssen. Infolgedessen sind gerade die Mitarbeiter verloren gegangen, auf die viel eingeredet wurde, die sich aber dennoch nicht an der Verschwörung beteiligt haben. Upps. Dieser Fehler wird nicht noch einmal passieren. Mit dem Wissen, dass der Kontakt-Graph ein gerichteter und kreisfreier Graph (DAG) ist, will Dr. Meta nun die Quelle allen Übels finden.

1. Beweise, dass ein DAG eine Quelle hat. (1 Punkt)
2. Gib einen Algorithmus im Pseudocode an, der einen DAG mit n Knoten und m Kanten als Eingabe erhält und in Zeit $O(n + m)$ alle Quellen ausgibt. Verwende dabei die folgende Signatur: (2 Punkte)

`LISTSOURCES($G = (V, E)$: Graph): List(Node)`

3. Angenommen, ein Algorithmus löscht iterativ Quellen aus einem DAG. Beschreibe welche Datenstruktur geeignet ist, um die Quellen zu verwalten, sodass eine Quelle v in Zeit $O(\deg(v))$ gelöscht werden kann. (2 Punkte)
Hinweis: Beim Löschen einer Quelle in einem DAG können neue Quellen entstehen.

Beim Nachdenken über DAGs hat Dr. Meta einen Einfall. Er vermutet, dass ein gerichteter Graph $G = (V, E)$ mit n Knoten genau dann azyklisch ist, wenn es eine topologische Sortierung gibt, also eine Reihenfolge v_1, \dots, v_n der Knoten, sodass für jede Kante $(v_i, v_j) \in E$ gilt, dass $i < j$.

4. Beweise Dr. Meta's Vermutung. (2 Punkte)
Hinweis: Achte darauf, dass hier zwei Richtungen gezeigt werden sollen.

Der Kontakt-Graph soll nun erneut ausgedünnt werden. Diesmal ist der Plan jedoch, die k Knoten zu erwischen, die den Verrat ins Rollen gebracht haben.

5. Gib einen Algorithmus in Pseudocode an, der als Eingabe einen DAG G mit n Knoten und m Kanten erhält und in Zeit $O(n + m)$ eine topologische Sortierung von G ausgibt. Verwende dabei die folgende Signatur. Das ausgegebene Array soll die Knoten in aufsteigender topologischer Sortierung enthalten. (3 Punkte)

TOPOSORT($G = (V, E) : \text{Graph}$) : [Node; n]

Hinweis: Verwende das Löschen eines Knotens v (in $O(\deg(v))$) als Subroutine.

Aufgabe 2 - xkcd#1323 (9+3 Punkte)

Alice besitzt n verschließbare, goldene Kisten und m goldene Schlüssel. Jede Kiste ist mit einem Schloss gesichert in das mindestens ein Schlüssel passt. Jeder Schlüssel passt in genau ein Schloss. Eine verschlossene Kiste kann nur auf zwei Arten geöffnet werden: entweder mit einem passenden Schlüssel oder destruktiv mit einem Hammer. Malory hat nun einige Schlüssel gestohlen und alle anderen in den Kisten verschlossen! Alice kennt die Zuordnung von Schlüsseln zu Schlössern und konnte immerhin beobachten welche Schlüssel in welcher Kiste gelandet sind. Da sowohl Schlüssel als auch Kisten wertvoll sind, möchte Alice möglichst wenige Kisten zerstören, um an alle Schlüssel zu gelangen.

1. Wie kann die obige Situation mithilfe eines gerichteten Graphen modelliert werden? (2 Punkte)
2. Gib einen Algorithmus im Pseudocode an, der deinen Graphen als Eingabe erhält und in Zeit $O(n + m)$ entweder einen Kreis ausgibt oder erkennt, dass der Graph keine Kreise enthält. (3 Punkte)

Wir nehmen nun an, dass die Schlüssel so in den Kisten verschlossen wurden, dass der resultierende Graph keine Kreise hat.

3. Alice möchte für eine gegebene Kiste wissen, ob sie alle Schlüssel bekommen kann, indem sie genau diese Kiste gewaltsam öffnet. Beschreibe wie sich Alice' Problem in deine Formulierung als Graphenproblem übersetzt und beschreibe einen Algorithmus, der das Problem in Linearzeit löst. (2 Punkte)
4. Eventuell genügt es nicht, nur eine Kiste mit dem Hammer zu öffnen. Alice fragt sich nun was die kleinste Anzahl Kisten ist, die zerschlagen werden müssen um an alle Schlüssel zu gelangen. Übersetze auch diese Fragestellung in die Modellierung als Graph. Wie kann die Anzahl in Zeit $O(n + m)$ bestimmt werden? (2 Punkte)

Im Folgenden nehmen wir nun nicht mehr an, dass der Graph kreisfrei ist.

- *. Beschreibe einen Algorithmus der nun bestimmt, wie viele Kisten mindestens zerschlagen werden müssen, um an alle Schlüssel zu gelangen. (3 Punkte)

Hier noch Werbung der Fachschaft:

Eulenfest am 14. Juli 2022

Nächste Woche Donnerstag findet das erste Eulenfest seit 2020 statt! Los geht es um 19 Uhr, im und um den Infobau. Dieses Jahr gibt es Livemusik von den Bands *Fancity* und *Sonnenblumen of Death*. Natürlich gibt es auch wieder DJs. Außerdem werdet ihr mit Getränken und Essen versorgt. Weitere Infos zum Fest findet ihr unter eulenfest-karlsruhe.de.

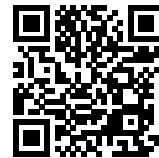


Gerne könnt ihr die Fachschaft mit einer Helferschicht unterstützen. Tragt euch dafür einfach selbst in das Helfersystem unter redseat.de/eulenfest22 ein.

Die Fachschaft sucht 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 Peter (peter.maucher@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



Helfersystem