Technische Universität Wien
Institut für Computergraphik und Algorithmen
Arbeitsbereich für Algorithmen und Komplexität

186.813 Algorithmen und Datenstrukturen 1 VU 6.0

Übungsblatt 2

für die Übung am Montag den 11. bzw. Dienstag den 12. April 2016.

Geben Sie bis **spätestens Sonntag, 10.4.2016, 23:59 Uhr** über TUWEL an, welche Beispiele Sie bearbeitet und gelöst haben. Gehen Sie dabei folgendermaßen vor:

- TUWEL (`https://tuwel.tuwien.ac.at`)
  Kurs *186.813 Algorithmen und Datenstrukturen 1 (VU 6.0)*
- Übungsblätter
- Bearbeitete Beispiele ankreuzen **und** abgeben
  - Link *Übungsblatt 2 - Details & Bewertung*
    Button *Abgabe bearbeiten*
    Bearbeitete Beispiele anhaken und *Änderungen speichern*.
  - Link *Hochladen Lösungen Übungsblatt 2*
    Button *Abgabe hinzufügen*
    PDF-Datei mit Lösungen hochladen und *Änderungen sichern*.

Bitte beachten Sie:

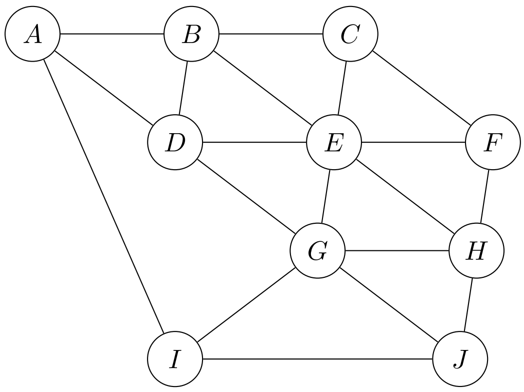
- Sie können **vor** der Deadline beliebig oft ihre Auswahl an Beispielen und das zugehörige Lösungs-PDF verändern, aber **nach** der Deadline gibt es **keine** Veränderung ihrer angekreuzten Beispiele **und** der PDF-Datei!
- Sie können Ihre Lösungen entweder direkt in einem Textverarbeitungsprogramm erstellen, oder aber auch gut leserliche Scans bzw. Fotos von handschriftlichen Ausarbeitungen einreichen.
- Bitte geben Sie Ihren Namen, Matrikelnummer und E-Mail-Adresse in den Ausarbeitungen an.
- Wenn Sie zur Präsentation Ihrer Lösung eines von Ihnen angekreuzten Beispiels ausgewählt werden und dieses aber nicht bearbeitet haben oder dieses Beispiel nicht in der PDF-Datei vorhanden ist, verlieren Sie **alle** Punkte dieser Übungseinheit!

  (Zusätzlich werden stichprobenartig die abgegebenen PDF-Dateien auf Übereinstimmung mit der entsprechenden Kreuzerlliste überprüft.)

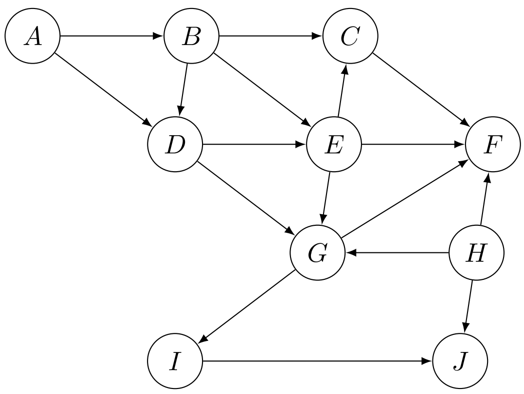
Aufgabe 9 Sortieren Sie das Array $A = [17, 24, 13, 5, 9]$ jeweils mittels Selection-Sort und Insertion-Sort entsprechend den Algorithmen aus der Vorlesung. Stellen Sie die einzelnen Zwischenschritte, die die Algorithmen ausführen, dar. Es genügt das Array nach jeder Iteration der äußersten Schleife abzubilden. Wie viele Vergleichsoperationen führt jeder der Algorithmen durch?

Aufgabe 10 Führen Sie auf nachfolgendem Graphen Breiten- und Tiefensuche entsprechend den Algorithmen aus den Vorlesungsfolien durch. Geben Sie dabei jeweils die Reihenfolge an, in der die Knoten besucht werden. Verwenden Sie jeweils A als Startknoten.

(Für die Abgabe genügt es, eine entsprechend gereihte Liste der Knoten anzugeben.)



Aufgabe 11 Bestimmen Sie für nachfolgenden gerichteten Graphen eine topologische Sortierung unter Verwendung des entsprechenden Algorithmus aus den Vorlesungsfolien.



Aufgabe 12 Gegeben sei ein schlichter ungerichteter Graph $G = (V, E)$. Finden Sie einen Algorithmus, der in linearer Laufzeit bestimmt, ob jede Zusammenhangskomponente von G aus einer geraden Anzahl an Knoten besteht. Entwickeln Sie eine Lösung in detailliertem Pseudocode und argumentieren Sie, dass der Algorithmus korrekt ist und die geforderte Laufzeitschranke gilt.

Aufgabe 13 Ermitteln Sie eine Darstellung des Graphen aus Aufgabe 10 als Adjazenzmatrix und als Adjazenzlisten.

Aufgabe 14 Für einen schlichten ungerichteten Graphen $G = (V, E)$ soll bestimmt werden, ob er einen Kreis der Länge 3 enthält.

- Überlegen Sie sich zuerst, wie das mit einem sehr einfachen Algorithmus in Zeit $O(n^3)$ durchgeführt werden kann. Eine grobe Skizze der Idee und die Argumentation der Korrektheit sowie der Laufzeitschranke sind hier ausreichend.
- Finden Sie nun einen besseren Algorithmus zur Lösung des Problems, dessen Laufzeit in $O(n \cdot m)$ liegt. Sie können annehmen, dass $n \leq m$ gilt. Entwickeln Sie eine Lösung in detailliertem Pseudocode und argumentieren Sie, dass der Algorithmus korrekt ist und die geforderte Laufzeitschranke gilt.

  Hinweis: Die angegebene Laufzeitschranke lässt es zu, zwischen der Darstellung als Adjazenzmatrix bzw. -listen zu transformieren.

Aufgabe 15 Wir haben in der Vorlesung den gerichteten Graphen G^{rev} kennengelernt. Bezogen auf einen gerichteten Ausgangsgraphen $G = (V, E)$ ist die Menge der Kanten von $G^{\text{rev}} = (V, E^{\text{rev}})$ durch $E^{\text{rev}} = \{(u, v) \mid (v, u) \in E\}$ gegeben. Finden Sie einen Algorithmus mit linearer Laufzeitgarantie, um für einen gerichteten Graphen G sein Gegenstück G^{rev} zu erzeugen. Verwenden Sie zur Repräsentation Adjazenzlisten. Entwickeln Sie eine Lösung in detailliertem Pseudocode und argumentieren Sie, dass der Algorithmus korrekt ist und die geforderte Laufzeitschranke gilt.

Aufgabe 16 Das Wiener Verkehrsmanagement plant alle Straßen in Einbahnen umzuwandeln. Es wird behauptet, dass es im neuen Straßennetz immer noch möglich ist, von jeder beliebigen Kreuzung jede andere Kreuzung zu erreichen. Die Gegner dieser Neuerung sind jedoch nicht überzeugt. Nun wird ein Algorithmus benötigt, um die Korrektheit der Aussage zu prüfen. Da die Änderungen schon bald in Kraft treten sollen, ist nur mehr wenig Zeit verfügbar und deshalb nur ein Algorithmus mit linearer Laufzeitgarantie anwendbar.

- Formulieren Sie das beschriebene Problem unter Verwendung von Graphen und erklären Sie, wie es in linearer Laufzeit gelöst werden kann.
- Nehmen Sie nun an, dass sich die ursprüngliche Behauptung als Irrtum herausgestellt hat. Stattdessen soll nun überprüft werden, ob die schwächere Behauptung, dass man von jeder Kreuzung die vom Rathaus erreicht werden kann, auch wieder zum Rathaus zurückkehren kann, gilt. Zeigen Sie, dass dies ebenfalls in linearer Laufzeit möglich ist.

(Eine verbale Beschreibung mit stichhaltiger Argumentation der Korrektheit und der Laufzeitschranke ist jeweils ausreichend. Sie dürfen Algorithmen und Resultate aus den Vorlesungsfolien benutzen.)
