RHEINISCHE FRIEDRICH-WILHELMS UNIVERSITÄT BONN
INSTITUT FÜR INFORMATIK DER UNIVERSITÄT BONN

Institut für Informatik 1
Computational Analytics
Prof. Dr. Petra Mutzel
M. Sc. Jonas Charfreitag
M. Sc. Philip Mayer

Übungen zur Vorlesung
Graphenalgorithmen
Sommersemester 2024

Übungsblatt 1
Besprechung:
23.4. und 25.04.

Bitte beachten Sie die Hinweise zu den Übungen und der Übungsgruppenverteilung im eCampus. Stellen Sie Fragen zu Vorlesung und Übung vorzugsweise im dort angelegten Forum.

Sie sind unbedingt dazu eingeladen, die Aufgaben in Gruppen zu bearbeiten. Um die Punkte für einen Aufgabenteil bekommen zu können müssen Sie natürlich in der Lage sein, die Aufgabe(n) allein und vollständig vorstellen zu können.

Aufgabe 1.1 – Kurzfragen (2 Punkte)

a) Beschreiben Sie kurz in eigenen Worten das stabile Heiratsproblem. Was ist gegeben, was ist gesucht?

b) Geben Sie an, wann eine Paarung bei dem stabilen Heiratsproblem instabil genannt wird.

c) Wozu wird die zweite Bedingung aus der *While*-Schleife vom Algorithmus von Gale & Shapley (*Liste von m nicht leer*) benötigt?

d) Beschreiben Sie ein Zuordnungsproblem (das keine Erwähnung in Vorlesung oder Übung findet), das nicht als stabiles Heiratsproblem modelliert werden kann.

Aufgabe 1.2 – Worst-Case Gale & Shapley (2 Punkte)

Im folgenden geht es um den Algorithmus von Gale & Shapley, wie er in der Vorlesung vorgestellt wurde.

a) Berechnen Sie die Größe des Lösungsraums für das stabile Heiratsproblem. Wie viele mögliche Paarungen gibt es?

b) Argumentieren Sie, warum im Algorithmus von Gale & Shapley höchstens $n^2 - n + 1$ Anträge gemacht werden können und geben Sie ein Beispiel für $n = 4$ an, für das diese Schranke erreicht wird.

Aufgabe 1.3 – Heiratsproblem - Variante (7 Punkte)

Die folgende Variante des Heiratsproblems findet in den USA bei der Zuordnung von Medizinstudierenden zu Krankenhäusern Anwendung. Das Verfahren ist wie folgt:

- Alle Studierende erstellen eine geordnete Liste der Krankenhäuser ihrer Wahl. Wir nehmen hier an, dass die Studierenden allen Krankenhäusern Prioritäten zuordnen müssen.
- Alle Krankenhäuser erstellen eine geordnete Liste aller Studierenden (auch wenn ein einzelnes Krankenhaus natürlich nicht genügend Plätze für alle Studierenden hat).
- Aufgrund dieser Listen erfolgt eine automatisierte Zuordnung der Studierenden zu den Krankenhäusern. Dabei wird davon ausgegangen, dass die Gesamtanzahl der zur Verfügung stehenden Plätze genau der Anzahl der Studierenden entspricht.
- Unter zwei möglichen Zuordnungen der Studierenden zu Krankenhäusern bevorzugt jedes Krankenhaus für sich die bezüglich seiner Prioritäten lexikographisch kleinste Auswahl.

a) Beschreiben Sie einen Algorithmus, der auf demjenigen von Gale und Shapley basiert. Die Rolle der „Männer“ wird hierbei von den Krankenhäusern übernommen.

b) Welche Eigenschaften hat Ihr Algorithmus bzgl.

   i) Vollständigkeit: Erhalten alle Studierende einen Platz oder kann es sein, dass manche „leer“ ausgehen?

   ii) Stabilität?

   iii) Eindeutigkeit?

   iv) „Männer“-Optimalität?

c) Angenommen ein Krankenhaus K mit eigener Prioritätenliste P kann einsehen welche Prioritätenlisten alle anderen Krankenhäuser und die Studenten haben. Kann das Krankenhaus K seine eigene Prioritätenliste so zu einer neuen Liste P' adaptieren, dass K bessere Studenten bezüglich P erhält wenn der Algorithmus mit P' statt P ausgeführt wird.

Skizzieren Sie für b) und c) einen Beweis, oder geben Sie ein Gegenbeispiel an.
a) -> 1.5 Punkte b) -> 4 Punkte, c) -> 1.5 Punkte

Aufgabe 1.4 – Roommate-Problem (2 Punkte)

$2n$ Studierende sollen in ein Studentenwohnheim aufgenommen werden. Diese sollen nun mit Hilfe von Prioritätenlisten zu *Doppelzimmern* zugeordnet werden. Anders als im Heiratsproblem haben alle Studierenden Prioritäten für alle anderen Studierenden und beliebige Studenten können ein Paar bilden. Geben Sie an, ob für dieses Problem immer eine stabile Lösung existiert.

Aufgabe 1.5 – Hauptspeicherrepräsentation von Graphen (2 Punkte)

Eine vereinfachte und verkürzte Formulierung des No-free-Lunch-Theorems lautet: "Es existiert kein universell gutes Verfahren zum Abstrahieren von Datensätzen".
Erklären Sie die Bedeutung dieses Theorems anhand der in der Vorlesung vorgestellten Möglichkeiten Graphen zu repräsentieren. Vergleichen Sie darauf basierend konkret zwei Hauptspeicherrepräsentationen für Graphen.

Aufgabe 1.6 – [Programmieraufgabe] Einlesen von Graphen (5 Punkte)

Schreiben Sie ein Programm, das ungerichtete Graphen im *Extendend Edgelist Format* (siehe unten) einliest und geben Sie den minimalen und den maximalen Knotengrad aus. Laden Sie Ihr Programm zur Evaluation als Lösung für die '1_REE' Aufgabe des Contests 'GA_SS24' auf unserer Competitive Programming Plattform hoch. Es empfiehlt sich unbedingt, die auf der Webseite einsehbaren Beispieleingaben lokal zu testen. Sollten Sie hierbei Probleme haben, hilft Ihnen möglicherweise dieses Tutorial. Machen Sie sich außerdem Gedanken über die theoretische Effizienz Ihrer Implementierung. Das Hochladen von ineffizientem Code wird mit hoher Wahrscheinlichkeit zu einem Überschreiten des auf der Plattform eingestellten 'Timelimits' führen.

Eingabe: Extended Edgelist Format Die erste Zeile der Eingabe enthält die Anzahl der Knoten $|V|$ und die Anzahl der Kanten $|E|$, durch ein Leerzeichen getrennt. Alle weiteren Zeilen enthalten jeweils eine Kante, repräsentiert durch die Knoten IDs der inzidenten Knoten. Die Knoten IDs sind aus $\{0, \dots, |V| - 1\}$.

Ausgabe Die geforderte Ausgabe für einen Eingabe-Graphen $G = (V, E)$ besteht aus einer Zeile in der erst der minimale Knotengrad des Graphen und dann durch ein Leerzeichen getrennt der maximal Knotengrad des Graphen steht. Also: '$d_{min}(V)$ $d_{max}(V)$'. Wobei $d_{min}(V) = \min\{d(v) : \forall v \in V\}$ und $d_{max}(V) = \max\{d(v) : \forall v \in V\}$.