



Algorithmische Mathematik I

Wintersemester 19/20

Prof. Dr. J. Gedicke

Johannes Rentrop und Jannik Schürg



Programmieraufgabenblatt 10. Abgabedatum: 07.01.2020–10.01.2020

Programmieraufgabe 1. (Bipartites Matching)

Bis zum Jahr 2030 ist die Zahl der Erstsemesterstudierenden so stark angestiegen, dass für die ALMa I Vorlesung 100 Tutorien eingerichtet werden müssen. Nachdem die benötigte Anzahl an TutorInnen eingestellt wurde, werden diese gebeten, anzugeben zu welchen Terminen sie können und zu welchen nicht. Dies ergibt einen bipartiten Graphen, bei dem die Knoten $0 \dots 99$ die TutorInnen darstellen und die Knoten $100 \dots 199$ die zu belegenden Tutoriumstermine. Eine Kante $\{12, 184\}$ würde z.B. bedeuten, dass Tutorin Nr. 12 potentiell Tutorium Nr. 84 wahrnehmen könnte. Sie finden den vollständigen Graphen auf der Vorlesungswebsite. Nun muss ein perfektes Matching gefunden werden!

Implementieren Sie dazu den Bipartiten Matching-Algorithmus (Algorithmus 8.42 aus der Vorlesung). Schreiben Sie ein Programm, das den Algorithmus auf den Eingabegraphen ausführt und das gefundene Matching ausgibt. Ist das gefundene maximale Matching perfekt?

Hinweis: Für die Verwaltung der vorkommenden Knotenmengen können Sie zum Beispiel `std::set` verwenden. Zum Auffinden eines Weges kann auf Code aus früheren Aufgaben zurückgegriffen werden.

(10 Punkte)

Programmieraufgabe 2. (Netzwerkflüsse)

- Implementieren Sie den Ford-Fulkerson-Algorithmus (Algorithmus 8.53 aus der Vorlesung), um einen maximalen Fluss in dem Flussnetzwerk, das auf der Vorlesungswebsite verfügbar ist, zu finden. Die Kantengewichte repräsentieren die Kapazitäten. Quelle und Senke sind durch die Knoten $n - 2$ und $n - 1$ gegeben, wobei n die Anzahl der Knoten ist.
- Man kann das Bipartite Matching-Problem leicht auf ein Maximalfluss-Problem zurückführen, indem man dem Graphen G mit Bipartition $V(G) = A \dot{\cup} B$ zwei neue Knoten s und t hinzufügt und s mit allen Knoten aus A verbindet, sowie alle Knoten aus B mit t verbindet (gerichtet). Alle bestehenden Kanten orientiert man von A nach B . Wie müssen nun die Kapazitäten gewählt werden? Wandeln Sie das Problem aus Aufgabe 1 in ein Netzwerkfluss-Problem um und verwenden Sie den Ford-Fulkerson-Algorithmus, um ein maximales Matching zu finden.

(10 Punkte)