Prof. Dr. A. Goerdt
Julian Pape-Lange

TU Chemnitz
Wintersemester 2021/2022
13.01.2022

Theoretische Informatik I

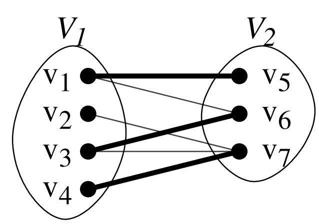
11. Übung

Abgabe: Lösen Sie die Aufgaben **1** und **3**. Ihre Lösungen geben Sie bitte entweder

- bis zum 19.01.2022 um 13:00 Uhr per Mail
  an `julian.pape-lange@informatik.tu-chemnitz.de`
  mit *Betreff:* `TI1 Hausaufgaben` oder
- bis zum 19.01.2022 um 13:00 Uhr im Briefkasten der Professur Theoretische Informatik (vor Raum A10.266.4)

ab.

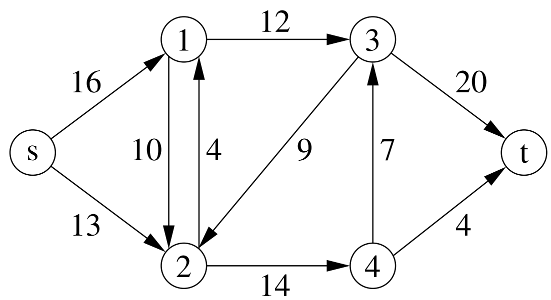
1. Aufgabe: (5P)
Ein *Matching* in einem ungerichteten Graphen $G = (V, E)$ ist eine Teilmenge von Kanten $M \subseteq E$, so dass gilt: Die Kanten aus der Teilmenge M haben keinen gemeinsamen Knoten. Ein Matching M hat die *maximale Größe*, wenn es kein Matching M' mit $|M'| > |M|$ gibt.



In bipartiten Graphen $G = (V_1 \cup V_2, E)$, wie im obigen Bild, läßt sich ein solches *Matching maximaler Größe* mit Hilfe des Algorithmus von *Ford-Fulkerson* bestimmen.
Geben Sie eine Konstruktion für ein Flussnetzwerk an, in dem der maximale Fluss der Größe eines maximalen Matchings entspricht.

2. Aufgabe:
Sei $G = (V_1 \cup V_2, E)$ ein bipartiter Graph, dessen Knotenmengen V_1 bzw. V_2 *Personen* bzw. *Jobs* darstellen. Eine Kante zwischen einer *Person* und einem *Job* symbolisiert, dass die Person die entsprechende Tätigkeit ausüben kann. Ziel ist es, soviele Jobs wie möglich abzudecken. Dabei können jeder Person bis zu *zwei* Tätigkeiten gleichzeitig zugemutet werden.
Lösen Sie das Problem mit Hilfe von Flussalgorithmen. Stellen Sie das entsprechende Netzwerk dar und erklären Sie, warum es geeignet ist.

3. Aufgabe: (5P)
Wir betrachten noch einmal den Graphen aus Übung 10:



Bestimmen Sie mit dem Algorithmus von Edmonds und Karp den maximalen Fluss.
Geben Sie außerdem einen Schnitt an, dessen Kapazität dem maximalen Fluss entspricht.

4. Aufgabe:
Geben Sie einen Graphen an, in dem es mehrere Schnitte mit minimaler Kapazität aber es nur einen Fluss maximaler Größe gibt.
Geben Sie auch an, welcher Schnitt im Beweis des Satzes *min cut - max flow* gefunden wird.