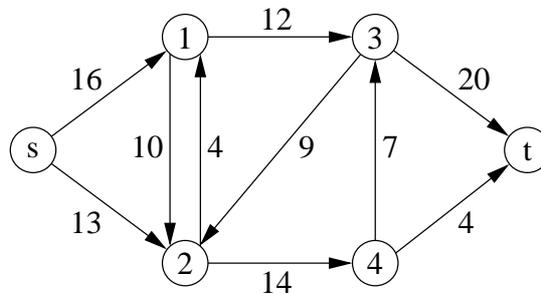


Theoretische Informatik I

10./11. Übung

Geben Sie die Lösung der Aufgabe 1 und 2 bitte bis zum 15.01.2018 9:15 Uhr ab. (Briefkasten vorm Raum 1/266 oder per eMail an falu@informatik.tu-chemnitz.de, *Betreff: TI1 Hausaufgaben*)

1. Aufgabe: Bestimmen Sie den *maximalen Fluss* durch das unten abgebildete Netzwerk. Nutzen Sie den Algorithmus von *Ford-Fulkerson* und geben Sie nach jeder Erhöhung des Flusses das *Restnetzwerk* und den *aktuellen Fluss* durch die Kanten an.



Gehen Sie davon aus, dass die Wege von s nach t in der folgenden Reihenfolge gefunden werden:

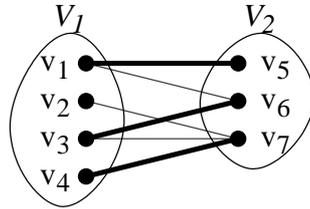
1. $(s, 1, 3, 2, 4, t)$
2. $(s, 2, 4, 3, t)$
3. $(s, 1, 3, t)$
4. $(s, 1, 2, 3, t)$

2. Aufgabe: Der Algorithmus von *Ford-Fulkerson* kann in eine Endlosschleife geraten, wenn das Flussnetzwerk *reelle* Kapazitäten besitzt. Zeigen Sie, dass *Ford-Fulkerson* in jedem Fall terminiert, wenn *rationale* Kapazitäten gegeben sind.

3. Aufgabe: Gegeben sei ein gerichteter Graph $G = (V, E)$ und Knoten $u, v \in V$. Weiterhin sei M eine Menge von *Wegen* vom Knoten u zum Knoten v , die jeweils *kantendisjunkt* zueinander sind.

Geben Sie einen Algorithmus an, der eine solche Menge M bestimmt. Die Größe der Menge M soll dabei *maximal* sein.

4. Aufgabe: 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.

5. 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, so viele 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.