Homework: The Max-Flow Problem

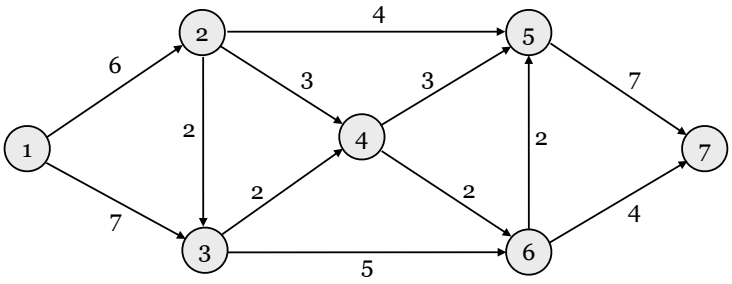
Note: Every homework has 10 credit points. To pass the homework, you need to reach 50% of the points. In this homework, only the first four exercises have credit points.

1. **(a)** What is meant by a feasible flow vector x? (0.5p)

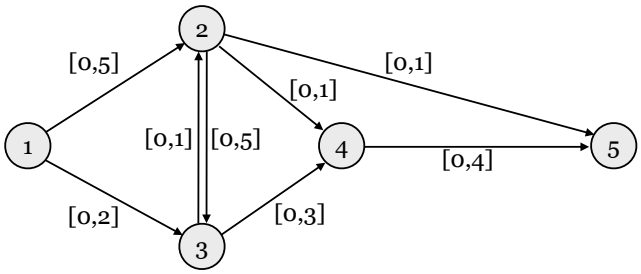
(b) If the lower flow bound for all arcs is 0, does there exist a feasible flow vector? If yes, suggest one or a method to find one. If no, state why. (1p)

(c) If the lower flow bound for all arcs is *not* 0, does there exist a feasible flow vector? If yes, suggest one or a method to find one. If no, state why. (1p)

2. Find the maximal flow from node 1 to node 7 in the following network. The numbers by the arcs are their upper flow bound, whereas their lower flow bound is assumed to be 0. (2.5p)



3. Consider the max-flow problem of the figure below, with $s = 1$ and $t = 5$.



(a) Enumerate all cuts of the form $[\mathcal{S}, \mathcal{N} - \mathcal{S}]$, such that $1 \in \mathcal{S}$ and $5 \notin \mathcal{S}$. Calculate the capacity of each cut. (1p)

(b) Find the maximal and minimal saturated cuts. (1p)

(c) Apply the Ford-Fulkerson method to find the maximum flow and verify the max-flow/min-cut equality. (0.5p)

4. Let i and j be two nodes of a directed graph $(\mathcal{N}, \mathcal{A})$.

(a) Consider the following algorithm, known as *breadth-first search* for finding a path from i to j. Let $T_0 = \{i\}$. For $k = 0, 1, 2, \ldots$, let

$$T_{k+1} = \{n \notin \cup_{p=0}^{k} T_p | \text{for some node } m \in T_k, (m,n) \text{ or } (n,m) \text{ is an arc}\},$$

and mark each node $n \in T_{k+1}$ with a label "$(m,n)''$ or "$(n,m)''$, where m is a node of T_k such that (m,n) or (n,m) is an arc respectively. The algorithm terminates if either (1) T_{k+1} is empty or (2) $j \in T_{k+1}$. If case (2) occurs, how would you use the labels to construct a path from i to j? (1p)

(b) Show that a path found by *breadth-first search* has a minimum number of arcs over all paths from i to j. (1p)

(c) Modify the algorithm of part (a) so that it finds a forward path from i to j. (0.5p)