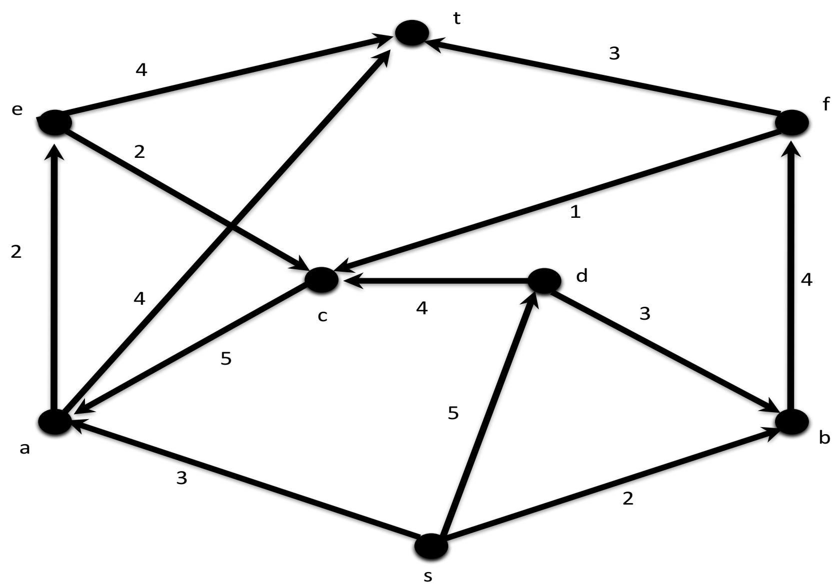
Algorithm Design Techniques *Due Tuesday February 4th*

Assignment 1: Network Flows

(1) *The Maximum Capacity Augmenting Path Algorithm.*
Consider the following maximum flow problem. (Arc capacities are shown.)

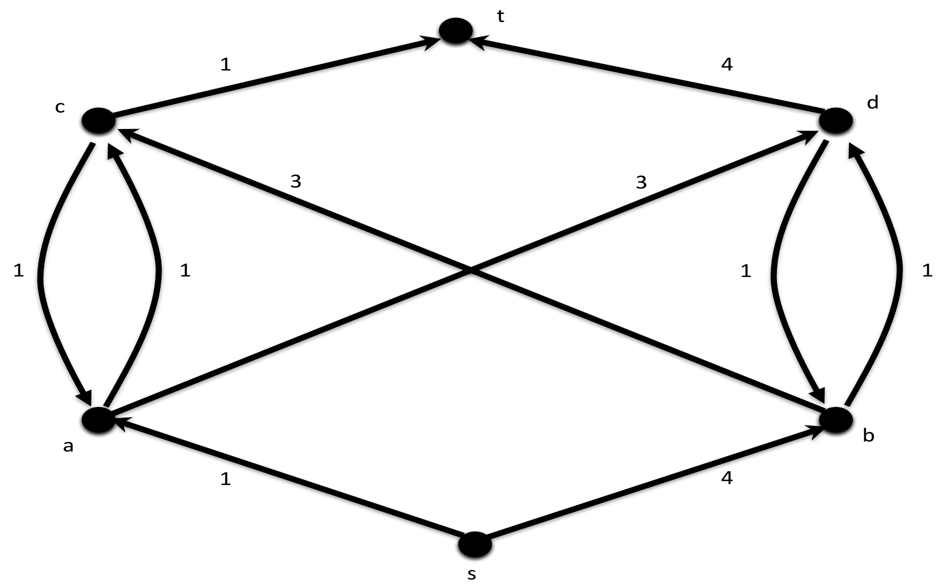


(a) Apply the Ford-Fulkerson algorithm to find a maximum $s-t$ flow. Use the **maximum capacity augmenting path method** to chose the augmenting path in each iteration. [In the case of a tie, break ties lexicographically; that is, select the augmenting path alphabetically.]

(b) Prove that your solutions give maximum $s-t$ flows by giving a certificate that shows it is impossible to find a larger flow.

(2) *Bottleneck Capacities: the Maximum Capacity Augmenting Path Algorithm.*
Suppose, in the tth iteration, the maximum capacity augmenting path algorithm uses a path with bottleneck capacity u_t in the residual graph $G_{\mathbf{f}_t}$. Is it always the case that the u_t are weakly decreasing, that is, $u_1 \geq u_2 \geq u_3 \geq \cdots$? Either prove this is true or present a counterexample.

(3) *The Shortest Augmenting Path Algorithm.*
Consider the following maximum flow problem. (Arc capacities are shown.)



(a) Apply the Ford-Fulkerson algorithm to find a maximum $s-t$ flow. Use the **shortest augmenting path method** to chose the augmenting path in each iteration. [In the case of a tie, break ties lexicographically; that is, select the augmenting path alphabetically.]
(b) Prove that your solutions give maximum $s-t$ flows by giving a certificate that shows it is impossible to find a larger flow.

(4) *Arcs in Cuts.*
Take a directed graph $G = (V, A)$ with arc capacity u_a for each arc $a \in A$. Let $\delta^+(S^*)$ be a minimum capacity $s-t$ cut in a graph $G = (V, A)$. Prove that if $(i, j) \in \delta^+(S^*)$ then there is no minimum capacity $s-t$ cut $\delta^+(\hat{S})$ such that $j \in \hat{S}$ and $i \in V \setminus \hat{S}$.

(5) *Cuts and Shortest Paths.*
Let $G = (V, A)$ be a directed graph with arc capacity $u_a = 1$ for each arc $a \in A$. Suppose that the shortest length path in G from s to t contains exactly k arcs. Prove that
(a) The maximum $s-t$ flow has value $O(\frac{m}{k})$.
(b) The minimum capacity $s-t$ cut has value $O(\frac{n^2}{k^2})$.

(6) *The Minimum Flow Problem.* Assume that each arc $(i, j) \in A$ has a lower bound $l_{ij} \geq 0$ and well as an upper bound u_{ij} on the amount of flow that must be routed along it. In the minimum flow problem we wish to send a flow f of minimum value from the source to the sink such that f satisfies the lower and upper bounds on every arc.

(a) Show how to solve the minimum flow problem using two applications of the max-flow algorithm on modified graphs with no lower bounds, i.e. all $l_{ij} = 0$.

(b) Prove the following minflow-maxcut theorem: let the floor of an $s-t$ cut $\delta^+(S)$ be $l(S) = \sum_{(i,j)\in\delta^+(S)} l_{ij} - \sum_{(i,j)\in\delta^-(S)} u_{ij}$. Show that the minimum value of all feasible $s-t$ flows equals the maximum floor of all $s-t$ cuts.

(7) [**Bonus Optional Question**]

Augmenting Paths with the Minimum Number of Backward Arcs.

Imagine that, in each iteration, the Ford-Fulkerson algorithm selects the augmenting path in G_f with the *fewest number of backward arcs.* Prove that:

(a) The total number of iterations is $O(mn)$.

(b) The total running time is $O(m^2n)$.