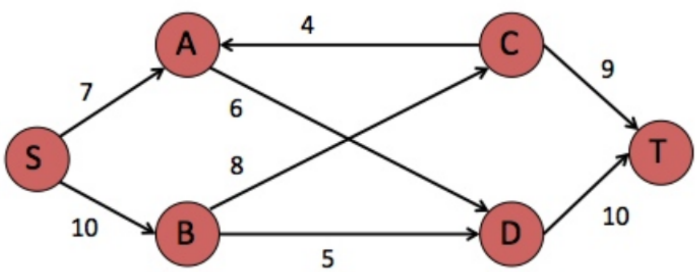
Advanced Algorithms — Exercise Set 1

Name: ____________________________

- Submit in class on **February 3, 2026**.
- Feel free to discuss with others, but write up your own solutions.
- Half points on this exercise set are awarded for completion / effort. Use it to learn!



Problem 1 (Go with the flow)**.**

(a) Find the maximum flow f for the graph above and a minimum cut. Don't just state the max-flow value, but also how much flow is sent along each edge.

(b) Draw the residual graph G_f at this maximum flow.

(c) An edge of a network is called a *bottleneck edge* if increasing its capacity results in an increase in the maximum flow (without changing any other capacities in the network). List all of the bottleneck edges in the above network.

(d) Give a simple example of a flow network which has no bottleneck edges.

Solution.

Problem 2 (Unit capacity networks)**.** State the running time of the Ford-Fulkerson algorithm that we showed in class. Now, consider the special case of max-flow in which every edge in the network has a capacity of 1. Explain why the running time of the Ford-Fulkerson algorithm is $O(mn)$ in this special case.

Solution.

Problem 3 (Bonus!)**.** In class, we showed that in every flow network with integer capacities, there is **some** max-flow which has all integer values. Give an example of a flow network with all integer capacities and a maximum flow on that network which has a non-integer value on at least one edge.

Solution.

Problem 4 (Multi-source / Multi-sink reduction)**.** In this problem we consider the generalization of the maximum flow problem with multiple source vertices $s_1, \ldots, s_k \in V$ and multiple sink vertices $t_1, \ldots, t_\ell \in V$. We still assume that no vertex is both a source and a sink, source vertices have no incoming edges, and sink vertices have no outgoing edges. A flow is defined as before: a nonnegative number f_e for each $e \in E$ such that capacity constraints hold on every edge and conservation holds at every vertex that is neither a source nor a sink.

The value of a flow is the total outgoing flow at the sources:

$$\sum_{i=1}^{k} \text{flow out of } s_i$$

Explain how to solve the maximum flow problem in graphs with multiple sources and sinks using the single-source single-sink version. (Hint: Consider adding additional vertices and/or edges.)

Solution.

Problem 5 (Global Minimum-cuts)**.** In the (undirected) global minimum cut problem, the input is an undirected graph $G = (V, E)$ with a nonnegative capacity c_e for each edge $e \in E$, and the goal is to identify a **global cut**, i.e., a partition of V into non-empty sets A and B that minimizes the total capacity $\sum_{e \in \delta(A)} c_e$ of the cut edges. (Here, $\delta(A)$ denotes the edges with exactly one endpoint in A.)

Explain why this problem reduces to solving $\binom{n}{2}$ maximum flow problems in undirected graphs. That is, given an instance of the global minimum cut problem, show how to

1. produce $\binom{n}{2}$ instances of the maximum flow problem (in undirected graphs) such that
2. given maximum flows to these $\binom{n}{2}$ instances, you can compute an optimal solution to the global minimum cut instance.

[Hint: remember that you can use max-flow to get the minimum capacity cut which separates any given pair of vertices]

Bonus: there is a way to do this with only $n - 1$ max-flow computations. Can you see how?

Solution.