For all algorithms, explain time complexity and prove correctness.

1. Draw out a maximum $s$–$t$ flow for the directed graph in the figure below as well as the corresponding residual graph. You only need to show the maximum flow, not the intermediate steps that led you to it. What is the minimum cut that corresponds to this maximum flow?

2. Let $G = (V, E)$ be a directed graph, $s \in V$ a source node and $t \in V$ a sink node. Also let $(S, \bar{S})$ be a minimum $s$–$t$ cut in the graph. Suppose we add 1 to the capacity of every edge in the graph. Must $(S, \bar{S})$ still be a minimum cut? If so, prove it; if not, give a counterexample.

3. Suppose we are given a directed graph $G = (V, E)$, a source $s \in V$ and sink $t \in V$, where instead of capacities on edges, each internal vertex $v$ has a capacity $c_v$ on the total flow that is allowed to pass through it. Each edge can carry an arbitrary amount of flow. Give an algorithm to find the maximum flow in such a network. Your algorithm should run in time polynomial in the number of nodes and edges. You may assume that the vertex capacities are positive integers. (Hint: try to convert the problem into a flow network of the type we are used to.)

4. Suppose you are given a directed graph $G = (V, E)$, a source $s \in V$, and a sink $t \in V$. This particular graph has unit capacities, i.e., $c_e = 1$ for every $e \in E$. You are also given a parameter $k \geq 0$. The goal is delete $k$ edges so as to reduce the maximum $s$–$t$ flow in $G$ by as much as possible. In other words, you want to find a set of edges $F \subseteq E$ so that $|F| = k$ and the maximum $s$–$t$ flow in $G' = (V, E\setminus F)$ is as small as possible subject to this. Give a polynomial-time algorithm for this problem.
5. Suppose you are given $n$ unit-length jobs, and each job $j$ has an integral release time $r_j$. Time is slotted, i.e., divided into intervals of the form $[t, t+1)$ for $t = 0, 1, 2, 3, \ldots$ so that a job can be scheduled entirely in one slot, but not across an integer boundary.

You want to find a schedule so that no job has to wait too long to be satisfied. Formally, if job $j$ is scheduled at slot $[t, t+1)$, we say that the \textit{wait time} of job $j$ is $t - r_j$. (So if a job is released at time 2, and then scheduled immediately in slot $[2, 3)$, it has a wait time of zero.) Give a polynomial time algorithm that finds a schedule that minimizes the maximum wait time of any job.