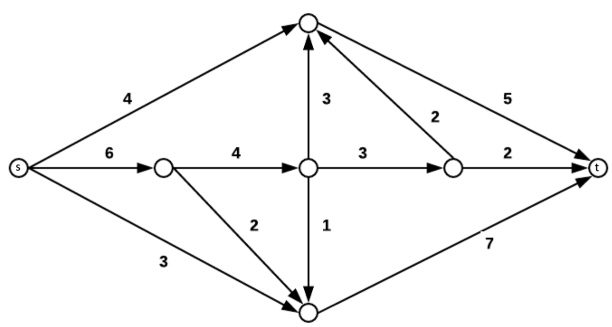
CS 577: Introduction to Algorithms	Homework 8
Out: 04/06/21	**Due: 04/13/21**

Name: ____________________ Wisc ID: ____________________

Ground Rules

- Answer the questions in the boxes provided on the question sheets. If you run out of room for an answer, add a page to the end of the document. Do **not** feel obligated to fill the entire solution box. The size of the box does **not** correspond to the intended solution length.
- The homework is to be done and submitted individually. You may discuss the homework with others in either section but you must write up the solution *on your own*.
- You are not allowed to consult any material outside of assigned textbooks and material the instructors post on the course websites. In particular, consulting the internet will be considered plagiarism and penalized appropriately.
- The homework is due at 11:59 PM CT on the due date. No extensions to the due date will be given under any circumstances.
- Homework must be submitted electronically on Gradescope.

Problem 1



(a) An edge in a flow network is called *upper-binding* if increasing its capacity by one unit increases the maximum flow in the network. Similarly, an edge in a flow network is called *lower-binding* if reducing its capacity by one unit decreases the maximum flow in the network. Identify all of the *upper-binding* and all of the *lower-binding* edges in the above flow network.

(b) Describe and analyze an algorithm for finding **all** of the upper-binding edges in a flow network G when given a maximum flow f^* in G. Your algorithm should run in time $O(n + m)$, where n is the number of nodes and m is the number of edges.

Problem 2

A given flow network G may have more than one minimum (s, t)-cut. While all minimum (s, t)-cuts will have the same capacity, they may have different numbers of edges directed from the s side to the t side. Let us define the **best** minimum (s, t)-cut to be any minimum cut with the smallest number of edges crossing the cut directed from the s side to the t side of the cut. (When the minimum cut is unique, by default it is also the best.)

(a) Describe and analyze an efficient algorithm to find the **best** minimum (s, t)-cut in a given flow network with integral capacities. (You may use as a subroutine an efficient max-flow algorithm without describing it in detail.)

(b) Describe a polynomial time algorithm to determine whether the best minimum cut is unique.