CS4800: Algorithms — S'18 — Jonathan Ullman

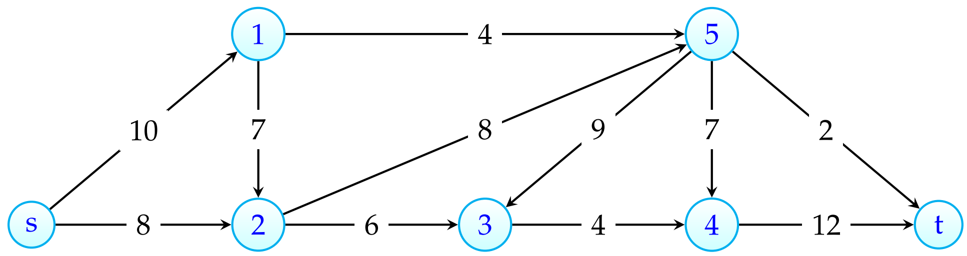
Homework 8
Due Friday Mar 23 at 11:59pm via Gradescope

Name:
Collaborators:

- Make sure to put your name on the first page. If you are using the LaTeX template we provided, then you can make sure it appears by filling in the `yourname` command.
- This assignment is due Friday Mar 23 at 11:59pm via Gradescope. No late assignments will be accepted. Make sure to submit something before the deadline.
- Solutions must be typeset in LaTeX. If you need to draw any diagrams, you may draw them by hand as long as they are embedded in the PDF. I recommend using the source file for this assignment to get started.
- I encourage you to work with your classmates on the homework problems. *If you do collaborate, you must write all solutions by yourself, in your own words.* Do not submit anything you cannot explain. Please list all your collaborators in your solution for each problem by filling in the `yourcollaborators` command.
- Finding solutions to homework problems on the web, or by asking students not enrolled in the class is strictly prohibited.

Problem 1. *Flow Practice*

Consider the following flow network $G = (V, E, \{c(e)\}, s, t)$.



(a) Compute a maximum s-t flow in this network and its value. State the sequence of augmenting paths used to obtain the flow.

Solution:

(b) Compute a minimum s-t cut in this network and its capacity.

Solution:

Problem 2. *Updating a Maximum Flow*

The military has built an elaborate shipping network to get vibranium from the Wakandan vibranium mines to their secret shield factory in upstate New York. The shipping network is represented as a flow network $G = (V, E, \{c(e)\}, s, t)$ with *integer* capacities. You have just finished computing an integer-valued maximum s-t flow f^* for this network when your boss comes in and says he is considering building some new shipping lines and demolishing others. You don't want to start over and compute a new flow from scratch, so you'll have to get creative.

For both problems below, clearly describe your algorithm, justify its correctness, and analyze its running time. Both algorithms should be substantially faster than computing a new maximum flow from scratch.[1]

(a) Design an algorithm that takes the original maximum s-t flow f^* and an edge e and finds the new maximum s-t flow f' that results from *increasing* the capacity of edge e by 1 (or, if $e \notin E$, the new maximum flow f' that results by inserting edge e with capacity $c(e) = 1$).

Solution:

(b) Design an $O(n+m)$ time algorithm that takes the original maximum s-t flow f^* and outputs a list of all edges $e \in E$ such that increasing the capacity of e by 1 increases the value of the maximum s-t flow.[2]

Solution:

(c) Design an algorithm that takes the original maximum s-t flow f^* and an edge $e \in E$ and finds the new maximum s-t flow f' that results from *decreasing* the capacity of edge e by 1.

Solution:

[1] **Hint:** Think about how changing the capacity of an edge changes the residual graph G_{f^*}.

[2] **Hint:** Running the algorithm from part (a) once for each edge in the graph will be too slow.