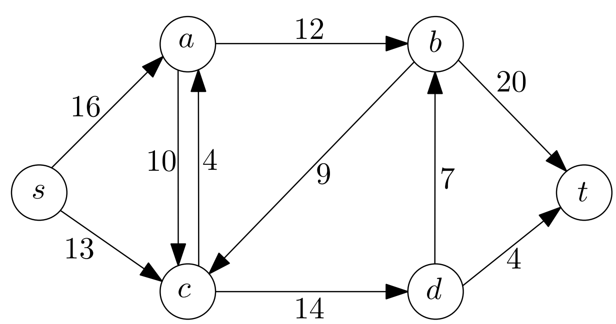
CS:5350 Homework 6, Spring 2016

Due in class on Thu, Mar 24

Collaboration: You are welcome to form groups of size 2 and work on your homeworks in groups. Of course, you are not required to work in groups. Every group should make one submission and names of both group members should appear on the submission and both students in a group will receive the same score. Other than the TA and the professor, you can only discuss homework problems with your group partner. Collaboration can be positive because talking to someone else about these problems can help to clarify your ideas and you will also (hopefully) get to hear about different ways of thinking about the problem. On the other hand, collaboration can be negative if one member of the group rides on work being done by the other member – please avoid this situation. If your solutions are (even partly) based on material other than what has been posted on the course website, you should clearly acknowledge your outside sources.
Late submissions: No late submissions are permitted. You will receive no points for your submission if your submission is not turned in at the beginning of class on the due date.
Evaluation: Your submissions will be evaluated on correctness *and* clarity. Correctness is of course crucial, but how clearly you communicate your ideas is also quite important.

1. Consider the directed graph below, with source s and target t identified.



   Recall that the Ford-Fulkerson proof of the Maxflow-Mincut theorem implies a simple algorithm to compute maximum flows. This is informally described at the beginning of Section 23.4 in Prof. Jeff Erickson's notes (bottom of Page 5). Note that this algorithm leaves unspecified which s-t path in the residual graph to use for augmenting the flow. Show the execution of the Ford-Fulkerson algorithm on the above graph. Specifically, start with the all-zero flow and then for each flow f that is produced by the algorithm, draw the residual graph G_f and identify an (arbitrary) s-t path in G_f the algorithm uses for augmenting f. So you will be drawing the above graph a bunch of times with flows of increasing values and you will also be drawing a bunch of corresponding residual graphs.

   When you have finished computing the maximum flow in the above graph, identify a cut whose capacity equals the value of the maximum flow.

2. Let f be a flow in a graph G and let $P := s = v_1, v_2, \ldots, v_p = t$ be a path from s to t in the residual graph G_f. A *bottleneck* edge in P is an edge $v_i \to v_{i+1}$, $1 \le i < p$, with minimum residual capacity. In Section 23.6.1 of Prof. Jeff Erickson's notes, there is description the Edmonds-Karp "fat pipe" rule that requires that we pick an s-t path in the residual graph whose bottleneck edge has greatest residual capacity and use this s-t path to augment the flow. This question is about how you would find such an s-t path in G_f. There is a two-sentence description of this algorithm in the second paragraph in Section 23.6.1:

   > Simply grow a directed spanning tree T, rooted at s. Repeatedly find the highest-capacity edge leaving T and add it to T, until T contains a path from s to t.

Note that in the above description "capacity" means "residual capacity" and the tree T is being grown using the residual graph G_f. Prove that the algorithm described above does indeed find an s-t path in G_f with a bottleneck edge with maximum residual capacity.

3. Problem 1 from the Chapter 23 Exercises, on Page 10.

4. Problem 9(a) from the Chapter 23 Exercises, on Page 11.
