Please hand in each problem on separate sheets with your name and netID on each. If a problem requires multiple sheets, please staple the sheets for that problem together.

**Reading:** Chapters 7.1 - 7.3, 7.5 and 7.7.

**Question 1**

This question refers to the figure on the back of this sheet. Please show your work on this figure and hand it in for your solution to Question 3.

At the top of the page you’ll find a flow network \( G \). We basically want you to simulate the main steps of the basic Ford Fulkerson algorithm and compute the maximum \( s - t \) flow in \( G \). In particular, you should specify the path \( P \) you pick to augment along, the resulting flow \( f \), and the resulting residual graph \( G_f \) at each step. As the handy sheet suggests, you should only need three iterations. Finally, you should report the value of the maximum flow, and describe a minimum cut.

Since the Ford Fulkerson algorithm is underspecified (in the sense that you can pick any path to augment along), we want you to choose paths using the following rule: among available augmenting paths, pick the one that comes first alphabetically given the labels used in the figure.

If, in the course of your work, you’ve erased and crossed out so much crap that it looks more like a Jackson Pollock painting than a worksheet on network flows, please be nice to the TAs (and yourself) and print out another copy from the course website.

**Question 2**

Let \( f \) be a flow in a network \( G = (V, E) \), and let \( \alpha \) be a real number. The scalar flow \( \alpha f \) is a function from \( E \) to \( \mathbb{R} \) defined by

\[
(\alpha f)(e) = \alpha \cdot f(e)
\]

Prove that the set of \( s - t \) flows in a network is a convex set by showing that if \( f_1 \) and \( f_2 \) are \( s - t \) flows, then for any \( \alpha \) in the range \( 0 \leq \alpha \leq 1 \), the function \( g = \alpha f_1 + (1 - \alpha) f_2 \) is also a flow. In addition, prove that \( v(g) = \alpha v(f_1) + (1 - \alpha) v(f_2) \).

**Question 3**

Recall that in class, when we defined a flow network \( G = (V, E) \) with source \( s \) and sink \( t \), we assumed that there were no edges into \( s \) and there were no edges out of \( t \). It turns out all the results we proved in class still hold without this assumption. The only difference is we would have needed a more general definition for \( v(f) \) (subtracting off any flow that comes back into \( s \)). In this problem, you will show that our assumption wasn’t actually all that restrictive. In particular, prove the following statement.

In any flow network \( G \) (which may have edges into \( s \) and out of \( t \)) for which the max flow has value \( k \), there exists an \( s - t \) flow \( f \) with \( v(f) = k \) that assigns 0 flow to all edges into \( s \) and all edges out of \( t \). That is, \( f \) maximum flow and sends no flow into \( s \) and no flow out of \( t \).