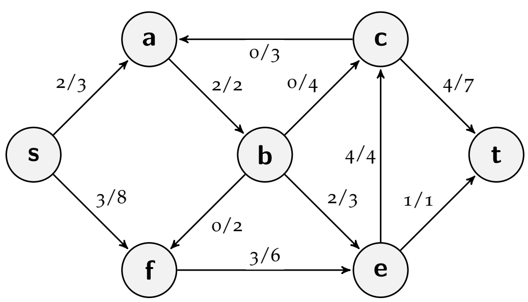

Collaboration Policy: You are encouraged to collaborate with up to 4 other students, but all work submitted must be your own *independently* written solution. List the computing ids of all of your collaborators in the `collabs` command at the top of the tex file. Do not share written notes, documents (including Google docs, Overleaf docs, discussion notes, PDFs), or code. Do not seek published or online solutions for any assignments. If you use any published or online resources (which may not include solutions) when completing this assignment, be sure to cite them. Do not submit a solution that you are unable to explain orally to a member of the course staff. Any solutions that share similar text/code will be considered in breach of this policy. Please refer to the syllabus for a complete description of the collaboration policy.

Collaborators: list your collaborators

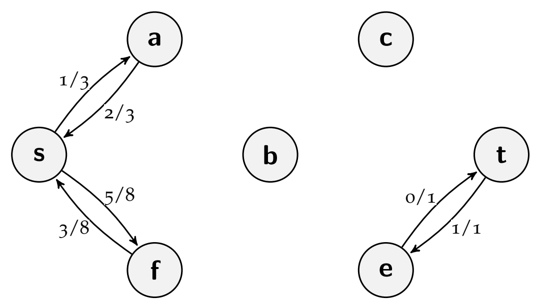
Sources: list your sources

PROBLEM 1 *Max Flow*

Given the following Flow Network G and the beginning of the Residual Graph G':



Flow Network G



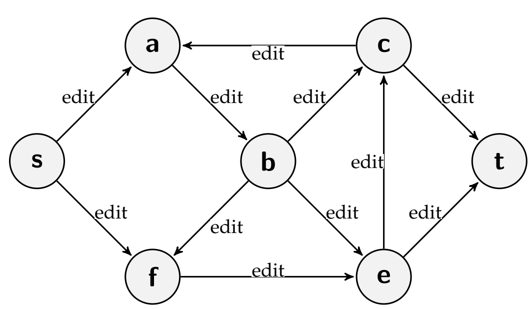
Residual Graph G'

1. Complete the Residual Graph G' above. You **must** edit the graph in LaTeX above (do not upload a picture).

2. Find an augmenting path in the graph G'. List the nodes in the path you found in order (e.g., $s \to a \to b \to c \to d \to t$).

   Solution:

3. Update the Flow Network G above. You **must** edit the graph below (do not upload a picture).



4. Find the min cut of the graph. List the nodes on each side of the cut.

   Solution:

PROBLEM 2 *Closest Pair of Points Reduction*

Reduce Element Uniqueness to Closest Pair of Points in $O(n)$ time. Element Uniqueness is defined as: given a list of numbers, return true if no number appears more than once (i.e., every number is distinct). Remember that closest Pair of Points is defined as: given a list of points (x, y), return the smallest distance between any two points.

Solution:

PROBLEM 3 *Insert/Delete/Min Data Structure*

Prove whether there exists a data structure where the operations `INSERT` (which inserts a given element into the data structure), `DELETE` (which removes a given element from the data structure, should it be present), and `MIN` (which returns the minimum element from the data structure) require $O(1)$ worst-case time each.

Solution: