

CSC 440

Assignment 6: Network Flow

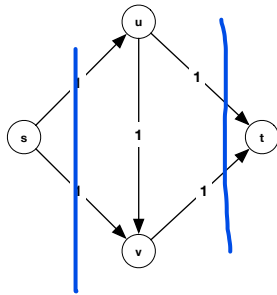
Due Wednesday, April 20th by 11:59PM

You may work in small groups on this assignment, but the work you hand in must be your own. You may submit this in one of two ways:

- Write it up electronically (e.g. LaTeX) and upload a PDF to Gradescope.
- Scan your handwritten solution (you can use a scanning app for a smartphone, such as Evernote's free Scannable app or the Notes app in iOS 11) and upload a PDF to Gradescope.
- Solutions handed in on paper **will not be accepted**

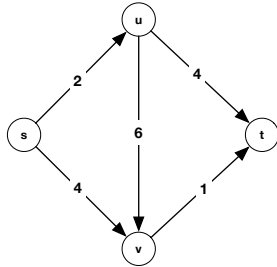
Your full name: **Jason Bachman**

1. (10 points) List all the minimum cuts in the following flow network (the edge capacities are the numbers on each edge)



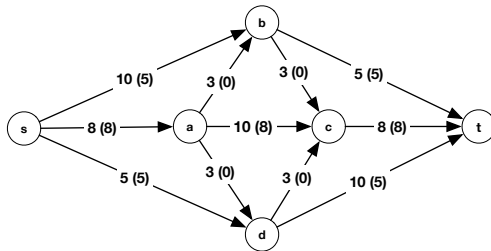
There are 2 min cuts as shown on the diagram.

2. (10 points) What is the minimum capacity of an (s, t) cut in the following network?



The minimum capacity of an s - t cut in this diagram is 5

3. (20 points) In the following network, a flow has been computed. The first number on each edge represents capacity, and the number in parentheses represents the flow on that edge.



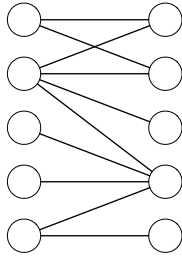
- a. (10 points) What is the value of the flow that has been computed? Is it a maximum (s, t) flow?

computed flow is 18. Not a maximum. $(a, b) = 8$, $(b, c) = 3$
 $(a, d) = 3$ $(d, t) = 8$ max flow is 21

- b. (10 points) Find a minimum (s, t) cut on the network, state which edges are cut, and state its capacity.

The minimum cut of the figure cuts edges (b, t) , (c, t) , (a, d) , and (s, d)

4. (25 points) At the beginning of the semester, you implemented an algorithm for finding a perfect matching based on a preference list on a complete bipartite graph. Here, consider a different problem: given a bipartite graph that isn't necessarily complete, determine the maximal matching (and if that matching is perfect). Note that here there are no preference lists; there are only vertices and edges. So, we are not worried about *stability*. Consider the following bipartite graph.



- a. (15 points) How might you use the Ford-Fulkerson Maximum Flow algorithm to determine whether or not there is a perfect matching? Hints: flow networks need edge capacities, so you'll need to add some. And, flow networks use directed edges, so you'll need to make this a directed graph. You may need some other changes as well.
- b. (10 points) On the example graph shown, what is the size of a maximum matching? Is it a perfect matching?

5. (25 points) Suppose we generalize the *max-flow* problem to have multiple source vertices $s_1, \dots, s_k \in V$ and sink vertices $t_1, \dots, t_l \in V$. Assume that no vertex is both a source and a sink, the source vertices have no incoming edges, and sink vertices have no outgoing edges, and that all edge capacities are still integral. A flow is still defined as a nonnegative integer f_e for each edge $e \in E$ such that capacity constraints are obeyed on every edge, and conservation constraints hold at all vertices that are neither sources nor sinks. The value of a flow is the total amount of outgoing flow at the sources $\sum_{i=1}^k \sum_{e \in \delta^+(s_i)} f_e$. Prove that *max-flow* with multiple sources and sinks can be reduced to the single-source single-sink version of the problem. Specifically, given an instance of multi-source multi-sink *max-flow*, show how to (i) produce a single-source single-sink instance that lets you (ii) recover a maximum flow of the original multi-source. Sketch out a proof of correctness, and that your algorithms run in linear time (not counting the time required to solve *max-flow* on the single-source single-sink instance). Hint: consider adding additional vertices or edges to the graph.

Space for problem 5

6. (10 points) Describe a real-world problem, not yet discussed in class, that is amenable to *max-flow* or *min-cut*. How would you represent the problem in terms of max-flow or min-cut? Given the various asymptotic complexities of the algorithms for *max-flow* discussed so far, which algorithm might be most appropriate, and why?