1. **Uniqueness, and lack thereof.**

   (a) Prove that *blocking flows* are not necessarily unique by finding an alternate blocking flow for the level graph in RET Figure 8.5(b).

   (b) Construct a counterexample (not necessarily the same graph as part (a)) to show that *maximum flows* are not necessarily unique.

2. **Human computation.** Simulate the execution of Sleator and Tarjan’s algorithm for computing a blocking flow on the following level graph. Show the result of each advance, retreat, and augment+delete (you may lump these two together).

3. **Implementation details.** In this question, you will explain some of the details of Sleator and Tarjan’s maxflow algorithm using link/cut trees.

   (a) Briefly describe how to represent a graph $G$ such that the following operations can be performed in constant time: obtaining a single incoming edge (returns null if none exists); obtaining a single outgoing edge (returns null if none exists); and deleting an edge $[u, v]$. Your representation should work on a pointer machine (i.e., no arrays). You are free to introduce whatever fields are necessary, but please be clear about your use of pointers and what fields are stored on each type of object.

   (b) Briefly describe how to construct the level graph, $L$, given a graph $G$ and a flow $f$. Use the representation described above. What is the complexity of your algorithm?

4. (Grads only.) **Minimum path cover.** CLRS 26-2. (If the problem in your book isn’t called “Minimum path cover,” it’s probably the wrong problem!)

5. (Grads only.) **Maximum flow by scaling.** CLRS 26-5. (If the problem in your book isn’t called “Maximum flow by scaling” it’s probably the wrong problem!)