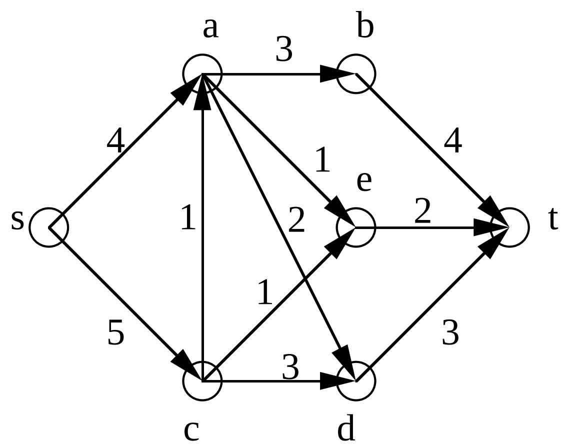
1. Find the maximum flow from s to t and the minimum cut between s and t in the network below. Show the residual network at the intermediate steps as you build the flow.



2. There are many variations on the maximum flow problem. For the two examples below, show how to solve the more general problem by reducing it to the original max-flow problem.

   - There are multiple sources and multiple sinks, and we wish to maximize the flow between all sources and all sinks.
   - Both the edges *and the vertices* (except for s and t) have capacities. The flow into and out of a vertex cannot exceed the capacity of the vertex.

   For the next problem, show how to reduce the problem to linear programming.

   - At each vertex, half the flow into the vertex is lost (or kept) at the vertex, and the other half flows out.

3. Consider the two-player game given by the following matrix. (A positive payoff goes to the row player.)

$$\begin{bmatrix} 5 & 1 & -2 & -3 \\ 8 & -4 & -3 & 1 \\ -3 & -1 & 5 & -2 \\ -9 & 6 & -4 & 9 \end{bmatrix}$$

   - Write down the linear program to determine the row player strategy that maximizes the value of the game to the row player. Do the same for the column player.
   - Find an LP solver on the World Wide Web. Use the solver to solve these linear programs, and give the proper strategies for both players.
   - What is the value of the game? Should the column player pay the row player to play, or vice versa, and how much should one player pay the other to make the game fair?

4. If we restrict the problems we look at, sometimes hard problems like counting the number of independent sets are in a graph become solvable. For instance, consider a graph that is a line on n vertices. (That is, the vertices are labelled 1 to n, and there is an edge from 1 to 2, 2 to 3, etc.) How many independent sets are there on a line graph? Also, how many independent sets are there on a cycle of n vertices?

Similarly, describe how you could quickly compute the number of independent sets on a complete binary tree. Calculate the number of independent sets on a complete binary tree with 127 nodes.

5. Consider the problem MAX-k-CUT, which is like the MAX CUT algorithm, except that we divide the vertices into k disjoint sets, and we want to maximize the number of edges between sets. Explain how to generalize both the randomized and the local search algorithms for MAX CUT to MAX-k-CUT and prove bounds on their performance.

6. We know that that all of NP-complete reduce to each other. It would be nice if this meant that an approximation for one NP-hard problem would lead to another. But this is not the case. Consider the case of Minimum Vertex Cover, for which we have a 2-approximation. We know (from the NP-completeness notes, which you may want to check) that C is a cover in a graph $G = (V,E)$ if and only if $V - C$ is an independent set in V. Explain why this does not yield an approximation algorithm for Maximum Independent Set. (Examples might be helpful.)

   The Maximum Independent Set problem and the Maximum Clique problem are related in the following way: an independent set of a graph G is a clique in the complement of G. (The complement of G is the graph that contains exactly the edges that are not in G.) Does an approximation algorithm for the Maximum Clique problem yield an approximation for Maximum Independent Set?