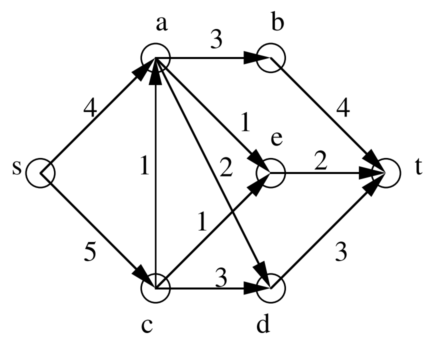
1. We have considered a random walk with a completely reflecting boundary at 0– that is, whenever position 0 is reached, with probability 1 we move to position 1 at the next turn. Consider now a random walk with a partially reflecting boundary at 0– whenever position 0 is reached, with probability 1/2 we move to position 1 at the next turn, and with probability 1/2 we stay at 0. Everywhere else the random walk either moves up or down 1, each with probability 1/2.

   Find the expected number of moves to reach n starting from position i using a random walk with a partially reflecting boundary. (You may assume there is a unique solution to this problem, as a function of n and i; you should prove that your solution satifies the appropriate recurrence.)

2. Find the maximum flow from s to t and the minimum cut between s and t in the network below, using the method of augmenting paths discussed in class. (This means give the flow along each edge, along with the final flow value; similarly, give the edges that cross the cut, along with the final cut value.) Show the residual network at the intermediate steps as you build the flow. (If you need more information on the algorithm, it's called the Ford-Fulkerson algorithm.)



3. You have been given a square plot of land that has been divided into n rows and columns, yielding n^2 square subplots. Some of these subplots have rocky ground and cannot support plant growth, while others have soil and can support growing a plam tree. You would like to plant palm trees on a subset of the square subplots so that every row and every column has exactly the same number p of palm trees. Furthermore, you would like to do this so that p is as large as possible. Devise an efficient algorithm to determine how to accomplish this. (You may given the running time in terms of the time to solve a suitable flow problem.)

4. Show how to reduce the following problems 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. The goal is to maximize the flow that reaches the destination t.
   - For each edge e, there is also a fixed cost c_e for each unit of flow through the edge. We need to find the maximum flow with the minimum cost. That is, there may be many possible flows that achieve the maximum flow; if there is more than one such flow, find the one of minimum cost. (Hint: you may need to use more than one linear program!)

5. Suppose we are given a maximum flow in a graph $G = (V,E)$ with source s, sink t, and integer capacities. (That is, we're given both the maximum flow value, as well as the amount of flow that goes along each edge to achieve that value.) Now the capacity of a given edge e is increased by 1. Give a linear time $O(|V| + |E|)$ algorithm for computing the new maximum flow. Similarly, give a linear time $O(|V| + |E|)$ algorithm for computing the new maximum flow if the capacity of a given edge e is decreased by 1.

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

$$\begin{bmatrix} 3 & 1 & 0 & -4 \\ 6 & -2 & -2 & 0 \\ -3 & -2 & 3 & -3 \\ -7 & 4 & -5 & 7 \end{bmatrix}$$

   - 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. 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?

7. **Do not turn this in. This is a suggested exercise.** For the Harvard Tutoring Club, there are n high schools students who need tutors, and when n available tutors in the club to tutor them. Each student needs a tutor, and each tutor can work with at most one student. Each high school student looks at the tutors and decides on a subset of them that they would want to work with. Suppose that for any subset S of the students, the collective set of tutors $T(S)$ that they are willing to work with satisfies the condition $|T(S)| \geq |S|$. Prove that there is a way to assign every student a tutor that they are willing to work with. Hint: think of this as a flow problem.