Homework 7: Max Flow Min Cut

Be sure to read the grading guidelines and style guidelines. Especially to see the suggested format for describing algorithms.

We sometimes describe how long are justifications or proofs are. These lengths are intended to help you estimate how much detail we're expecting, you should not take those estimates as hard length-limitations.

Our solutions for any individual problem will fit in approximately one page or less. **If you submit an answer substantially longer than 2 pages, the TAs are allowed to stop reading at the end of page 2.**

You are allowed (and encouraged!!) to collaborate with each other. Brainstorming is much easier to do in a group than alone! But you must follow the collaboration policy (which includes needing to write your submission on your own).

You will submit to gradescope; we have a different box for each problem, please give yourself time to submit.

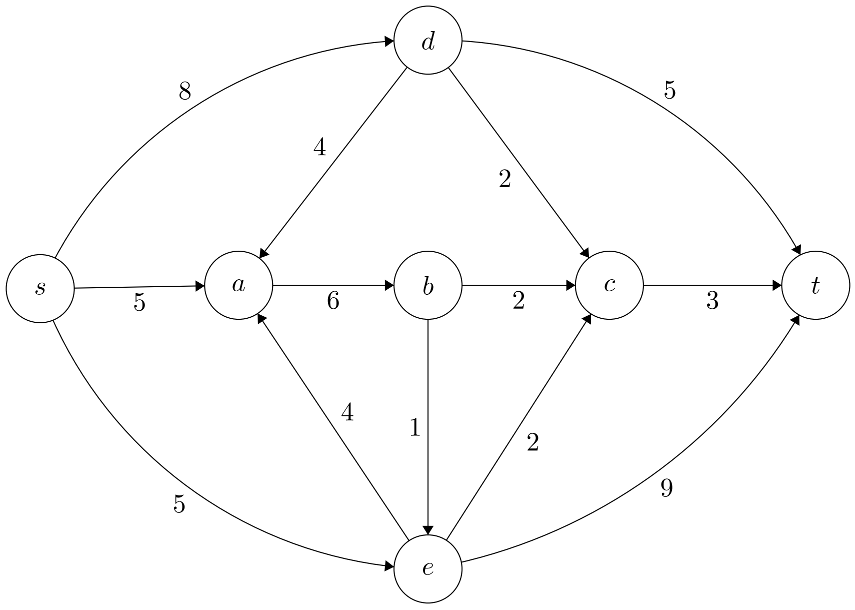
Note on running times
When analyzing running times on this homework, you may assume that creating a graph with n vertices and m edges takes $\mathcal{O}(n+m)$ time, as long as it is constant time to determine whether an edge exists (e.g., because you're reading it from input, or using a very simple rule to decide if edges exist).

1. Try the algorithm by yourself! [10 points]

Execute the Ford-Fulkerson algorithm on the following graph, showing the residual graph after each iteration.
At the end, give:

- the maximum flow (i.e., the amount of flow on each edge).
- The value of the maximum flow.
- The minimum cut (this should be formatted as two sets of vertices)
- The capacity of the minimum cut.

You might find this website useful if you are type-setting your work.



2. Release The Kraken

In class, we saw how to tell if the Mariners could still win the division. In this problem, we're going to see if the Kraken can still win the President's Trophy. The President's Trophy is awarded to the hockey team that finishes the season with the best record. Unlike in baseball, where every game ends in a win or a loss and the team with the most wins is the champion, hockey uses "points" to decide on standings. A hockey game can end in three different ways[1]:

- In regulation: the winner of the game gets 2 points in the standings and the loser gets 0.
- In overtime: the winner of the game gets 2 points in the standings, the loser gets 1 point in the standings.
- As a draw: both teams get 1 point in the standings.

Note that the number of points awarded per game can vary! Either 2 or 3 points total could be awarded.

The team with the most points in the standings wins the President's Trophy. Unlike in class, we want to see if the Kraken can win the trophy "outright" – it is not enough to have as many points as all other teams, we want them to have *strictly* more points than all other teams.

You'll be given:

- A list of n teams (including the Kraken), and their current number of points.
- A list of k games remaining to be played.

You should return `true` if the Kraken can still finish the season with strictly more points than any other teams and `false` otherwise.

(a) Describe a graph you can run a max-flow algorithm on. Be sure to mention edge directions and capacities.

(b) Briefly justify correctness. We aren't expecting a formal proof here, but you should have a sentence or two for each of the restrictions given in the problem.

(c) Describe how you'll tell whether an assignment is possible or not. If an assignment is possible, how do you read it from the maximum flow?

(d) Describe the runtime of the algorithm. Briefly justify why the running time is the value that you state.

3. Max-Chess Min-Box

As part of CSE 421 staff team-building, we decide to compete in a number of chess-boxing tournaments.[2] We have n staff members who are planning to attend t chess-boxing tournaments. In addition, we know that every staff member is either better at chess or at boxing and in the input we will designate them as a "chess-player" or a "boxer". As part of the training process, Robbie, a boxer[3], wants to ensure that there is at least one chess-player and at least one boxer representing our staff at every tournament. Finally, every tournament requires exactly p competitors and no staff member may participate in more than 10 tournaments (but may participate in any number from 0 to 9), and some staff members may be unavailable for some tournaments (tough grading load that week) and during these days you may not assign that staff member to a tournament.

[1]Observant hockey fans may notice the information below is quite out-of-date (as the third option was eliminated with the introduction of shootouts in the mid '00's). Consider this problem a historical analysis.

[2]Learn more about chess-boxing, our staff's favorite hobby, here.

[3]This designation reflects his lack of chess skill more than the existence of boxing skill.

More formally, you'll be given:

- A list of n TA's. For each TA:
  - whether they're a chess-player or a boxer
  - what tournaments they're unavailable for (these tournaments are numbered 0 through $t-1$)
- An integer t determining how many tournaments there are
- An integer p determining how many participants there must be at every tournament

Determine how to construct some assignment of staff to tournaments or determine no such assignment exists.

(a) Describe a graph you can run a max-flow algorithm on. Be sure to mention edge directions and capacities.

(b) Briefly justify correctness. We aren't expecting a formal proof here, but you should have a sentence or two for each of the restrictions given in the problem.

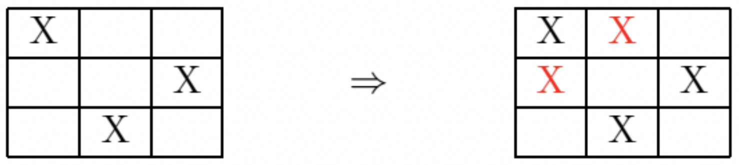
(c) Describe how you'll tell whether an assignment is possible or not. If an assignment is possible, how do you read it from the maximum flow?

(d) Describe the runtime of the algorithm. Briefly justify why the running time is the value that you state.

4. Wait, This is a flow problem?

Two of your friends are playing a game on a $n \times n$ chess board where they are trying to place 2-square $\times$ 1-square dominoes on the board. The game begins with an arbitrary set of tiles marked with an X. No domino can be placed on any tile marked with an X. You (feeling spiteful that you were not invited to play) wish to X out additional tiles on their chess board so that the game becomes unplayable, by which we mean there is no position at which a domino can be legally placed. However, you have a limited amount of time before they come to check back on their game, so you need to draw the fewest number of X marks possible such that no domino can be placed on the resulting board.

One example is shown on the table on the left, where before you walk up to the board 3 squares were already removed. Your algorithm should output 2, corresponding to the two squares marked with the red X on the right.



Note: Dominos are 2×1 dimensions, and can be of either vertical or horizontal orientation.

(a) Give an efficient algorithm that outputs the minimum number of extra tiles that we would need to cross out to guarantee that we cannot place any domino on the remaining board. (You must use a max-flow/min-cut-based algorithm in this problem).

(b) Explain why your algorithm works. We don't need a full proof here, but you should have an explanation of why the object you're finding in the last part corresponds to squares X'd out, and why the one you find gives you the minimum number of additional squares.

(c) Describe the runtime of the algorithm above. Let the board be $n \times n$, let the initial board have s squares that are already marked with an X, and suppose you find the answer is a additional squares must be marked. Briefly justify why the running time is correct.