Lab 0

Due February 22, 2021 before class

Cassandra, Shashank, Sherrie, and Manu were starting to get bored because they didn't have enough Advanced Algorithms work, so they decided they needed to find themselves a hobby. They found a local badminton league and each joined separate teams. Everyone on the team with the most wins at the end of the season will win a Dunkin Donuts gift card. Shashank has stated that once his team can no longer win the most games, he is going to quit. Therefore he wants to keep track of when his team has been mathematically eliminated from the competition.

Part 1 - Setting up the problem

The current standings are as follows:

				Against			
Team	**Wins**	**Losses**	**Left**	**Sherrie**	**Shashank**	**Manu**	**Cassandra**
Sherrie	83	71	8	-	1	6	1
Shashank	80	79	3	1	-	0	2
Manu	78	78	6	6	0	-	0
Cassandra	77	82	3	1	2	0	-

1) Which teams have been eliminated from getting the Dunkin Donuts prize? Which teams have not been eliminated? Why or why not?

Team Cassandra has been eliminated, because even if her team wins all the left 3 games, the most win could be 80 which is even less than the current win of team Sherrie. Shashank is also eliminated because he could win at most 83 games but there are 6 games left between Sherrie and Manu so one of them will always exceed 83. Sherrie and Manu both stand a chance.

2) Sherrie decides that she's going to be smarter than the rest of the Advanced Algorithms team and create an easy way to tell who's been eliminated. Due to bad record-keeping, Sherrie has access to none of the scores. However, she does have a 5 minute window to look at the standings to quickly determine what teams are eliminated. She decides she is going to set this up as a network flow problem.

The games won will be represented by w_{name} and the games remaining will be represented by r_{name}. For instance, Sherrie has won $w_{Sherrie}$ games and has $r_{Sherrie}$ games remaining. The teaching team trusts you can figure out the variable representation for the other players.

She sets it up as the following network flow:

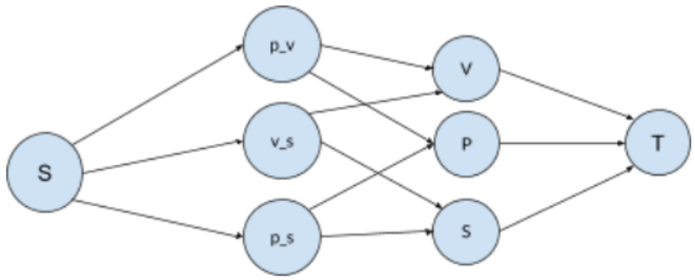

Figure 1: Sherrie's network flow

Figure 1 denotes the network flow diagram that Sherrie constructed to figure out if she was eliminated (note: the flow diagram is specific to her), without any of the capacities. The first node is a source node. The second series of nodes represent the matches between each of the other teams (i.e. Cassandra vs Shashank, Cassandra vs Manu). The third series of nodes represent the other teams (i.e. Cassandra, Shashank). The final node is a sink node.

Construct a procedure for determining if teams are eliminated. Note that to do this, you will have to determine what the capacities of the graph depicted in Figure 1 are. For this question, you must:

1. Draw out the graph with the capacities represented in variable form (explain what the variables represent).
2. Identify what the values of the variables would be for this specific problem
3. Write out the strategy for solving the problem.
4. Explain why this strategy works.

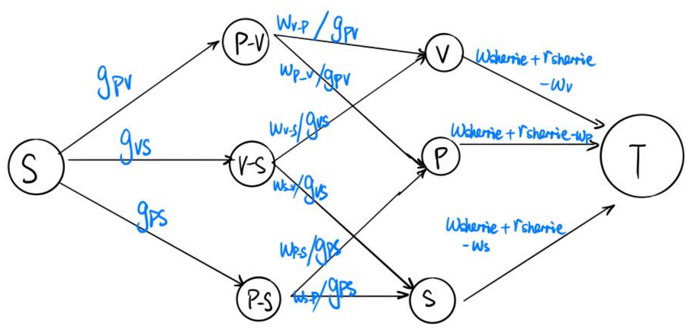

Figure 2: Network flow setup

let P be team Shashank, V be team Cassandra, and S be team Manu. Therefore from source to matches between teams we have g_{PV}, g_{VS} and g_{PS} each representing the game left between teams. The second series of edges represent the number of rest of games they win and therefore their capacities are equal to the numbers of games rest. The last series of edges are the games that team could still win.

Given the standings we have,

$$
\begin{aligned}
g_{PV} &= 2 \\
g_{VS} &= 0 \\
g_{PS} &= 0 \\
Cassandra: w_{Sherrie} + r_{Sherrie} - w_V &= 83 + 8 - 77 = 14 \\
Shashank: w_{Sherrie} + r_{Sherrie} - w_P &= 83 + 8 - 80 = 11 \\
Manu: w_{Sherrie} + r_{Sherrie} - w_S &= 83 + 8 - 78 = 13
\end{aligned}
$$

If the flow is feasible, then team Sherrie is eliminated. This works because if team Sherrie can win at most $w_{Sherrie} + r_{Sherrie}$ The other team, X, win $w_X + rwin_X$ the second series of edges represents the winning count of each team, so if $rwin_X \geq w_{Sherrie} + r_{Sherrie} - w_x$ then they should win or tie team Sherrie.

3) For the network flow diagram you finished above:

1. Convert it into a linear program (using variables, not the values). If you aren't sure how to do this, check out this link: http://www.mathcs.emory.edu/ cheung/Courses/323/Syllabus/NetFlow/max-flow-lp.html.
2. Provide an explanation of why this formulation makes sense, given the original context.

$$
\begin{aligned}
g_{PV} &= w_{V-P} + w_{P-V} \\
g_{VS} &= w_{V-S} + w_{S-V} \\
g_{PV} &= w_{P-S} + w_{S-P} \\
g_{PV} &\leq 2 \\
g_{VS} &\leq 0 \\
g_{PS} &\leq 0 \\
w_{V-P} + w_{V-S} &= w_{Sherrie} + w_{Sherrie} - w_V \\
w_{P-V} + w_{P-S} &= w_{Sherrie} + w_{Sherrie} - w_P \\
w_{S-P} + w_{S-V} &= w_{Sherrie} + w_{Sherrie} - w_S
\end{aligned}
$$

We are already given that the second series of nodes represents the games left between the two players. Therefore g_{PV} represents the game left between p and v, w_{P-V} means the number of games team p won team v. The last edge represents how many games this team need to win to eliminate Sherrie. Therefore we just need to check if any flow is feasible. If so, team Sherrie is eliminated by that team.

Part 2 - Implementation

Implement the network flows and the linear programming approach to the problem in Python (we are providing input files and starter code).

Make a fork of this github repo: https://github.com/AdvancedAlgorithms/Lab0.

Use "pip install -r requirements.txt" to install the requirements for the right libraries (you might want to use pip3 to use python3).

We have also provided a test file (test_badminton_elimination.py). At a minimum, your code should pass all of the tests in that file. Feel free to add your own additional test cases if you would like to more robustly test your code. If you think the test cases we have given you are sufficient, please explain how either in a comment or in your answer to this question. We aren't evaluating you on this test cases portion, but it's a good exercise to go through. To run your code on a specific input table (defined in a txt file, see teams2.txt and teams4.txt for examples), you can simply run "python badminton_elimination.py teams2.txt"

We recommend using the networkx function to solve the problem using network flows (documentation can be found here: https://networkx.github.io/documentation/networkx-1.10/reference/generated/networkx.algorithms.flow.maximum_flow.html) and using the picos solver to solve the problem using linear programming (documentation can be found here: https://picos-api.gitlab.io/picos/graphs.htmlmax-flow-min-cut-lp).

Your program should be able to answer the following question: Who is eliminated given a table of the current standings? You should be able to do this using a network flows approach and a linear programming approach.

Example input (the 4 at the top represents the # of teams in the division and the remainder of the rows and columns correspond to the same rows and columns as were specified in the table above):

4
Sherrie 83 71 8 0 1 6 1
Shashank 80 79 3 1 0 0 2
Manu 78 78 6 6 0 0 0
Cassandra 77 82 3 1 2 0 0

Corresponding output:
Sherrie: Eliminated? False
Shashank: Eliminated? True
Manu: Eliminated? False
Cassandra: Eliminated? True

To submit this lab - submit a link on Canvas to your Github repository with code and answers to questions 1-3.

Happy coding!