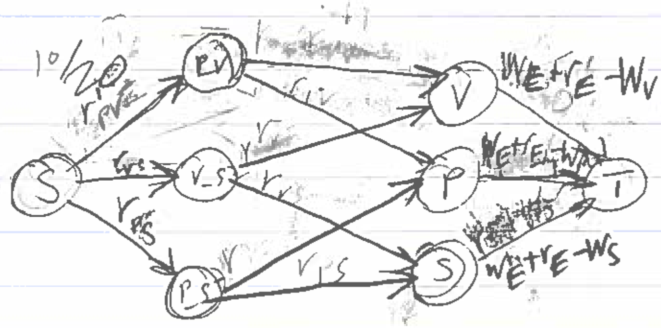
Pizza
cheesy
cheesy
cheesepizza

Pokemon version

Lab 0

2014

1. Vicky's team has already lost because Emily already has 83 games, and she can only win a maximum number of 80 games with 77 wins and 3 left. Prava can still not be eliminated since she can match Emily by winning 3 games and having 83 wins, with 80 games won and 3 leftover. Shashank has not been elimited because he can win a ~~maximum~~ of 84 games beating Emily. Emily can still win because she currently has the most wins and 8 games left to play

2. 

r_{pv} = # of games remaining between Prava and Vicky

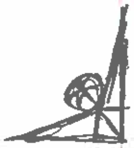
r_{vs} = # of games remaining between Vicky and Shashank

r_{ps} = # of games remaining between Prava and Shashank

W_E = The number of games Emily has won, r_E = The number of games Emily has left to play. W_V = The number of games Vicky has won. W_P = The number of games Prava has won. W_S = the number of games that Shashank has won

In order to solve the problem using this graph, you would find the max flow through the graph. The last three edges represent the maximum number of games someone would need to win to tie with Emily. If that edge is at capacity, then that team has at least tied with Emily. Then, if one of the first three edges (which represent the number of games between each team) and it leads to the at capacity final edges Emily has been eliminated.

The first 3 edges must be filled since they represent the remaining games between each team. If the edge at the end is full then the team has played and won only enough games to tie with Emily. However if the final edge is full, and the initial edges are not full, then Emily has been eliminated. Since the first 3 edges have



Lab 0

to be full, this configuration is impossible. If this impossible configuration is the only way for the team to not surpass Emily's wins, then Emily must have been eliminated.

3. max: $SP\_S + SV\_S + SP\_V$

Given: $SP\_V \leq r_{PV}$, $SV\_S \leq r_{VS}$, $SP\_S \leq r_{PS}$, $VT \leq W_E + r_E - W_V$,
$PT \leq W_E - r_E - W_P$, $ST \leq W_E - r_E - W_S$, $VT == P\_VV + V\_SVS$,
$PT == P\_VP + P\_SP$, $ST == V\_SS + P\_SS$, $SP\_S == P\_SP + P\_SS$,
$SV\_S == V\_SV + V\_SS$, $SP\_V == P\_VP + P\_VV$

The objective function is the sum of all the edges coming out of the start. This makes sense because these edges represent the games played between each team. We are trying to maximize the games played because the solution is only viable if each team plays all of their remaining games. The constraints represent various rules like team cannot play more than their remaining games or that the flow into a node must equal the flow coming out of it. There are also inequalities representing that the final flows going into the sink should not go above the number of games Emily has won, in order to find out if she has been eliminated, for the same reason as in the previous question.

Part 2 - Implementation

I did not add any additional test cases because I believe the current set of test cases already test a wide array of values and graph sizes. I could make test cases of more extreme sizes, but I believe that would test the efficiency of my graph generator more than my solver.