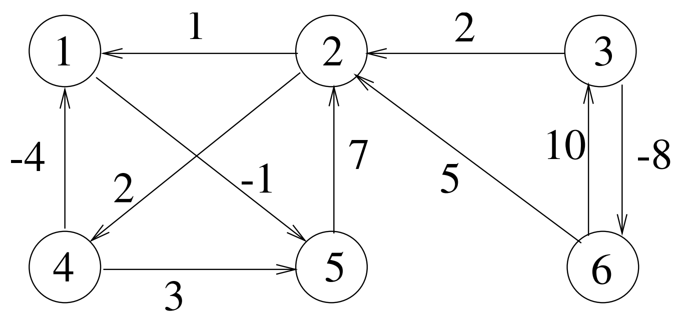
COMP 3711 – Spring 2019
Tutorial 10

1. Run the Floyd-Warshall algorithm on the weighted, directed graph shown in the figure. Show the matrix $D^{(k)}$ that results for each iteration of the outer loop.



2. (CLRS) Give an algorithm that takes as input a directed graph with positive edge weights, and returns the cost of the shortest cycle in the graph (if the graph is acyclic, it should say so).

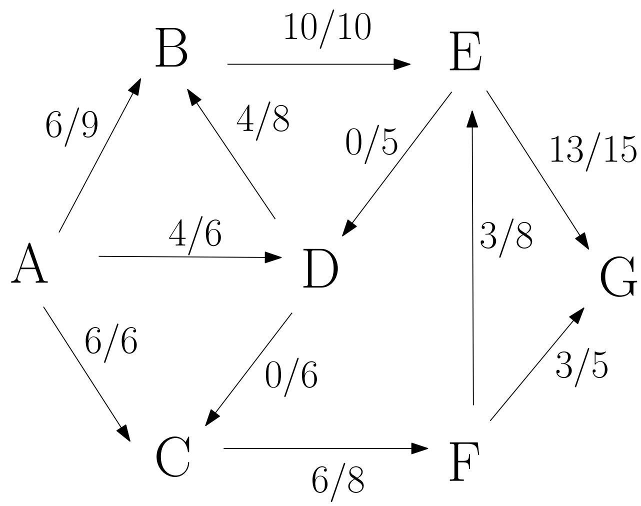
   Your algorithm should take time at most $O(n^3)$, where n is the number of vertices in the graph.

3. In the class notes on the Floyd-Warshall Algorithm we said that it was possible to reduce the space requirement from $O(n^3)$ to $O(n^2)$ by not keeping each of the n $n \times n$ matrices $D^{(i)}$ but instead keeping only ONE matrix and reusing it.

   We then wrote the code for doing that.

   Why does this space-reduced code work and give the correct answer?

4. Consider the given graph with flow values f and capacities c (f/c) as shown. $s = A$ and $t = G$.



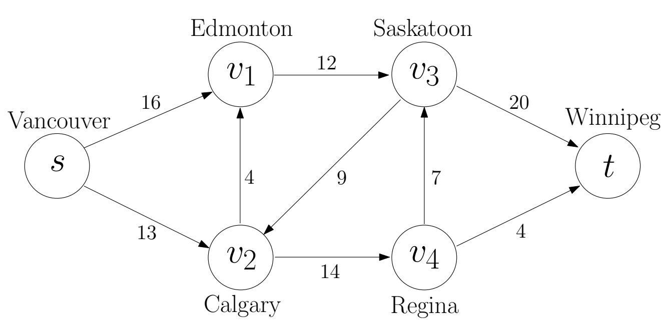
Draw, the residual graph.
Find an augmenting path.
Show the new flow created by adding the augmenting path flow.
Is your new flow optimal?
Prove or disprove.

5. Max-Flow as taught in class assumed a single source and single sink.

   Suppose the flow network has multiple sources $s_1, s_2, \ldots, s_m$ and multiple sinks $t_1, t_2, \ldots, t_n$ and the goal is to move as much from all the sources to all of the sinks as possible.

   Extend the flow properties and definitions to the multiple-source, multiple-sink problem. Show that any flow in a multiple-source, multiple-sink flow network corresponds to a flow of identical value in the single-source, single-sink network obtained by adding a supersource and a supersink, and vice versa.

6. Show the execution of the Edmonds-Karp algorithm on the following flow network.



Recall that the Edmonds-Karp algorithm is to implement Ford-Fulkerson by always choosing a *shortest path* (by number of edges) in the current residual graph.