CS 331: Algorithms and Complexity (Fall 2016)
Unique numbers: 51420, 51425, 51430, 51435

Assignment 6

Due on Wednesday, October 26, by 11.59pm

Matt Gmitro 51430 MTG759

Problem 1

(12 points) Let $G = (V, E)$ be a flow network, with a source s, a sink t, and a positive integer capacity c_e on every edge e. State whether each of the following is True or False. If True, you have to justify your answer. If False, you have to provide a counter example and show your work.

1. (3 points) Let $Q \subset V$, such that Q does not contain either of s or t. Then, $f^{out}(Q) - f^{in}(Q) = 0$.
   Answer: True because Q can only have flow coming from the source and going out to the target. Thus, the flow out must be equal to the flow in by the conservation condition, since all nodes in Q with flow going into them will pass on an equal amount of flow leaving them.

2. (3 points) If f is a maximum s-t flow in G, then f saturates every edge out of s with flow (i.e., for all edges e out of s, we have $f(e) = c_e$).
   Answer: False because the bottleneck may be less than the capacity of an edge out of s. For example, $(s) - 2 - > (u) - 1 - > (v) - 2 - > (t)$ where s, u, v, and t are nodes and the numerical values are the edge capacities.

3. (3 points) If all capacities are odd, then there is a maximal s-t flow f such that $f(e)$ is odd for every e.
   Answer: False. If the capacities for the edges out of the source sum to an even number then the amount going into t can be even. For example the maximal flow should be two in the below graph:

```
     1          1            3
 S+------>a+------->u+------------>t
 +                  ^
1|                  |
 v                  |
 b+------------------
          1
```

4. (3 points)For every G and every maximum s-t flow on G, there always exists an edge such that increasing the capacity on that edge will increase the maximum flow that's possible in the graph.
   Answer: False, for example, $S--1--->U---1-->V---1--->T$. Where (S,U,V,T) are vertices and all edges have capacity 1. Raising the capacity of any edge in this graph does not change the maximum possible flow.

Problem 2

(10 points) Given a directed graph, $G(V, E)$, which has integer edge weights, but contains only positive cost cycles, give an algorithm to efficiently find the number of shortest paths between two points, $s, t \in V$. You do **not** have to prove the correctness of your algorithm.

Answer:

```
ShortestPathCounter(G,s,t)
   n = number of nodes in G
   Array M[v, 0...n-1]
   Define M[0,t] = 0 and M[0,v] = infinity for all other v in V
   For i = 1, ..., n-1
      For v in V in any order
         M[i,v] = min(M[i-1,v],min(M[i-1,w]+Cvw))
   Let count = 0
   for i = 0, ..., n-1
      if (M[i,s] == M[n-1,s])
         count++
   return count
```

Problem 3

(13 points) Lee and Sal share an apartment and one of their classes. Unfortunately, they have recently fighted and stopped talking to each other. They are still living in the same apartment but refusing to go to school together. They even refuse to walk on any block that the other has stepped on that day. However, Lee and Sal have no problem with their paths crossing at a corner. The apartment and the school are on corners but they are not sure that they can go to the same school anymore. They provided you with a map of the their town. Show how can you get Lee and Sal to the same school as a maximum-flow problem (i.e., Translate the problem into a maximum-flow problem and then explain why/how it is a maximum flow problem). For simplicity, you can assume that each street or block can be traveled in only one direction which is marked on the map you are given.

Answer: To formulate the problem, we will start by forming a graph G = (V,E). The apartment will be the source node s and the school will be the target node t. Each corner represents a vertex in the set of vertices V. Each street is represented by an edge in set E. Let C be the set of capacities for each edge. Let f(u,v) where (u,v) is an edge in E be the flow on that edge. We let c(u ,v) be the capacity at an edge (u, v) and we set the capacity on each node to 1. The flow can thus only be 0 or 1 because flows are integer values. If a flow is passing over an edge, then Lee or Sal have walked on that edge and the other can not walk on it. If it is possible for both Lee and Sal to go to school, the flow into t will be at least 2 because they will be taking different paths from s in order to have multiple flows going into t. Thus, we can use a maximal flow algorithm to determine whether or not the maximum flow is at least 2. If it isn't at least 2, then it won't work for Sal and Lee to attend the same school.