CS 4820, Spring 2018	**Homework 6, Problem 2**
Name: Rongguang Wang	
NetID: rw564	
Collaborators: Siyuao Liu (sl2928); Yihao Chen (yc2288)	

(2) *(10 points)* In a flow network, let us define an edge e to be *useless* if the relation $f(e) = 0$ is satisfied by **every maximum flow** f. Design a polynomial-time algorithm that takes a flow network G and a maximum flow $\bar{f}$ on this network, and outputs a list of all of its useless edges.

For full credit, your algorithm's running time should be $O(m^2)$ or faster.

Solution:

The problem is aiming to find all useless edges, edges that are not used by any maximum flows, knowing multiple maximum flows may exist in the flow network G with a given maximum flow $\bar{f}$. There are few assumptions for this problem. Firstly, each node has at least one incident edge which means no isolated nodes exist. Secondly, no edges entering the source s and no edges leaving the sink t of the flow network G. Thirdly, the capacities of the edges are all integers. Here are some observations. To start with, if the flow value of an edge e in the given maximum flow $\bar{f}$ is larger than zero ($\bar{f}(e) > 0$), then this edge e is not an useless edge since it has been used for this particular flow $\bar{f}$. Thus, the useless edges can only be found within the subset of edges whose flow value in the given maximum flow $\bar{f}$ are equal to zero ($\bar{f}(e) = 0$). Then, if there exists a forward edge $\overrightarrow{e}$ from node u to node v and a backward edge $\overleftarrow{e}$ from node v to node u between any given nodes u and v, the flow value weight for each edges can be altered when keeping the maximum flow value. Thus, the edge (u, v) contains a cycle and both the forward edge and the backward edge of (u, v) are not useless, since the weight distribution of the flow value for the two edges may be different for different possible maximum flow f with the maximum flow value $\bar{f}$ unchanged. Therefore, if there exist a cycle between two nodes, the edges between the two nodes are not useless. We can apply the two observations into the algorithm to solve the problem.

Algorithm:

```
given f_bar(e) for any edge e in flow network G

if f_bar(e) > 0,

  the subset of edges e are not useless

else if f_bar(e) = 0,

  for each edges e whose end nodes are u and v:

    perform Depth First Search (DFS) at node u

      if the search hits u back,

        then a cycle was detected and all the edges in the cycle are not useless

      else
```

```
        no cycle was found and record the edge e as useless
  end for

output useless edges
```

Proof of Correctness:

Lemma 1. *Any edge is not useless if there exists a cycle between the two ends of the edge(s) with flow value zero.*

Proof. Each flow can be represented as a linear combination of paths and cycles and the difference between different maximum flows are paths and cycles. Since all maximum flows in a graph has the same flow value and the cycles result in no flow value changes while paths change the flow value, different possible maximum flows are only differ in cycles but not paths. Thus, by adding or subtracting cycles a given maximum flow can be transformed to another one. Therefore, the edges within any cycles of the flow network are not useless. □

Running Time:

There are m edges and n vertices in the flow network G. The cost of iterating the given maximum flow $\bar{f}$ is $O(m)$. It takes $O(m)$ time to iterate the edges in the for loop and $O(m+n)$ time are needed for the DFS within each iteration. Therefore, the total time complexity is $O(m + m \cdot (m+n))$ which is $O(m^2)$ since $m > n$.