Introduction to Algorithms	**Solution Set 6**
CS 4820, Spring 2018	

(1) *Computing a maximum flow in a network is a fairly time-consuming task, but various related problems have* linear-time *algorithms. Design linear-time algorithms for each of the following problems in a flow network with n vertices and m edges. Prove the correctness of your algorithm and prove that its running time is $O(m+n)$. In both problems, you are allowed to assume that the edge capacities are integers.*

(1a) Max-flow detection. (5 points)
Given a flow network G and a flow f in G, decide whether f is a maximum flow.

(1b) Flow improvement. (5 points)
Given a flow network G, and a flow f_0 in G that is not *a maximum flow, find another flow f_1 such that $v(f_1) > v(f_0)$.*

Solution.
(1a) Form the residual graph G_f and use BFS or DFS to search for a path from s to t in G_f. If no path is found, then f is a maximum flow. Otherwise, f is not a maximum flow. Forming G_f takes $O(m+n)$ time, as does the graph traversal (BFS or DFS) to test for existence of a path from s to t. If no path is found, then f satisfies the termination condition for the Ford-Fulkerson algorithm, and the proof that this implies it is a maximum flow was given in lecture. On the other hand, if a path P from s to t is found in G_f, then augmenting f using P produces a flow of strictly greater value, which proves that f is not a maximum flow.

(1b) Form the residual graph G_{f_0}, use BFS or DFS to search for a path P from s to t in G_{f_0}, and augment f_0 using P to obtain another flow f_1. It takes $O(m+n)$ time to build G_f, $O(m+n)$ to find the path P, and $O(n)$ to augment f_0 using P. This produces a flow f_1 whose value is

$$v(f_1) = v(f_0) + \text{bottleneck}(f, P) > v(f_0),$$

as desired.

(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. Compute the residual graph $G_{\bar{f}}$. (Running time $O(m)$.) For every edge e, if $f(e) = 0$ or e belongs to a directed cycle in G_f, then e is not useless. All other edges are useless.

Testing for a directed cycle in $G_{\bar{f}}$ containing any given edge $e = (u, v)$ takes linear time, using BFS or DFS to try finding a path from v to u. So, finding all edges that belong to directed cycles in $G_{\bar{f}}$ takes $O(m^2)$. (It's actually possible to design a linear-time algorithm that finds all edges belonging to directed cycles in $G_{\bar{f}}$. This is an easy consequence of the fact that strongly connected components of a directed graph can be computed in linear time. But any algorithm with running time $O(m^2)$ is a valid solution to this problem, because it didn't ask for a running time faster than $O(m^2)$.)

To prove correctness, we need to show that our criterion for uselessness is valid. Since $\bar{f}$ is a max-flow, if $\bar{f}(e) > 0$ then clearly e is not useless. If $\bar{f}(e) = 0$ but $G_{\bar{f}}$ contains a directed cycle through e, then we can augment this directed cycle (increase flow on forward edges and decrease flow on backward edges) to obtain another max-flow f' with $f'(e) > 0$.

Now we prove the converse: if e is not useless, then either $\bar{f}(e) > 0$ or e belongs to a directed cycle in $G_{\bar{f}}$. Let W be the set of all vertices reachable from v in $G_{\bar{f}}$, including v itself. Let $E(W, V - W)$ and $E(V - W, W)$ denote the set of all edges from W to $V - W$ and the set of all edges from $V - W$ to W, respectively. By our definition of W, we know that $G_{\bar{f}}$ has no edges of positive residual capacity leaving W, hence

$$\forall e \in E(W, V - W) \quad \bar{f}(e) = c_e \tag{1}$$
$$\forall e \in E(V - W, W) \quad \bar{f}(e) = 0 \tag{2}$$

For convenience define

$$\chi_s = \begin{cases} 1 & \text{if } s \in W \\ 0 & \text{otherwise} \end{cases}$$

$$\chi_t = \begin{cases} 1 & \text{if } t \in W \\ 0 & \text{otherwise.} \end{cases}$$

Every flow f'' satisfies

$$\begin{aligned} \sum_{e \in E(W, V-W)} c_e &\geq \left(\sum_{e \in E(W, V-W)} f''(e) \right) - \left(\sum_{e \in E(V-W, W)} f''(e) \right) \\ &= \sum_{w \in W} \left(\sum_{e \text{ out of } w} f''(e) - \sum_{e \text{ into } w} f''(e) \right) \\ &= v(f'') \cdot (\chi_s - \chi_t), \end{aligned} \tag{3}$$

and that the left side equals the right side if and only if the first inequality is tight, which happens if and only if f'' saturates every edge of $E(W, V - W)$ and sends no flow on every edge of $E(V - W, W)$. We know from (1)-(2) that this is exactly what happens when $f'' = f$.

Assuming $e = (u, v)$ is not useless, there exists a max-flow f' such that $f'(e) > 0$. Since $v(f') = v(f)$, the left and right sides of (3) are equal when $f'' = f'$. From this, we may conclude that f' saturates every edge of $E(W, V - W)$ and sends no flow on every edge of $E(V - W, W)$. However, since $f'(e) > 0$, this means that e does not belong to $E(V - W, W)$. As $v \in W$, we may conclude now that $u \in W$, implying that there is a path in $G_{\bar{f}}$ from v to u. Thus, either e itself is in $G_{\bar{f}}$, in which case it belongs to a cycle in $G_{\bar{f}}$, or e is not in $G_{\bar{f}}$, in which case $\bar{f}(e) = c_e > 0$. We have shown that if e is not useless, then either e belongs to a cycle in $G_{\bar{f}}$ or $\bar{f}(e) > 0$, which is what we wanted to prove.

Remarks. Solving this problem requires surmounting two obstacles: designing the algorithm, and proving its correctness. Designing the algorithm requires understanding the relationship between flows and residual graphs, which involves thinking about questions such as: if f is a max-flow and $f(e) = 0$, what ways can we imagine modifying f to make $f(e) > 0$ without decreasing the value of the flow? How does the residual graph $G_{\bar{f}}$ guide the search for these modifications?

Once the algorithm has been discovered, it is fairly challenging to come up with a complete proof of correctness. I think the best suggestion is to think about $\bar{f}, f'$ and the difference between them. You can represent $f' - \bar{f}$ as a circulation (i.e., a flow that satisfies flow conservation at *every* vertex, including the source and sink) and assuming $\bar{f}(e) = 0$ this circulation sends a positive amount of flow on edge e. From that description of $f' - \bar{f}$ you can try to show that e belongs to a cycle in $f' - \bar{f}$. Then it's not hard to argue that every edge of that cycle has positive residual capacity in $G_{\bar{f}}$, and that gives you the directed cycle in $G_{\bar{f}}$ that you were looking for.