CS 4820, Spring 2018 **Homework 6, Problem 1**
Name: Rongguang Wang
NetID: rw564
Collaborators: Siyuao Liu (sl2928); Yihao Chen (yc2288)

(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.

Solution:

The problem is aiming to design an algorithm which can be used to detect whether a given flow f in a flow network G is a maximum flow. Basically, Ford Fulkerson Algorithm can be used to solve this problem. Firstly, the residual graph G_f should be build for the given flow f. The residual capacity of each edge should be computed. Then, check whether there exists an augmenting path P, a path from source s to sink t in the residual graph G_f of the flow network G, by using Depth First Search (DFS). If there is no augmenting path P, then the given flow f is a maximum flow according to the termination condition of the Ford Fulkerson Algorithm and we output "yes". Otherwise, the given flow f can be augmented by the augmenting path P to flow f with a higher flow value and we output "no".

Algorithm:

```
given f(e) for any e

compute the residual graph G_f

if G_f contains an augmenting path P

  return no

else

  return yes
```

Proof of Correctness:

Lemma 1. *If f is a flow with no augmenting path P, then f is a maximum flow.*

Proof. Proof was given in the lecture. □

Lemma 2. *If f is a flow whose residual graph G_f contains an augmenting path P with bottleneck capacity b, then augmenting path P yields a flow f whose value is $v(f)+b$. Especially, the value of this* *flow f is greater than the value of f, so f is not a maximum flow.*

Proof. Proof was given in the lecture. $\square$

Running Time:

There are m edges and n vertices in the flow network G. The build of the residual graph takes $O(m+n)$ time. It takes $O(m)$ time to check whether the residual graph G_f contains augmenting path P by using DFS. In total, the time complexity is $O(m+n)$.

(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:

The problem is aiming to design an algorithm which can be used to improve a given flow f_0 to a flow f_1 which has a higher flow value ($v(f_1) > v(f_0)$) in a flow network G. The given flow f_0 is known as not a maximum flow for the flow network G. Basically, Ford Fulkerson Algorithm can be used to solve this problem. Firstly, the residual graph G_f should be build for the given flow f_0 and the residual capacity of each edge should be computed. Then, find an augmenting path P, a path from source s to sink t in the residual graph G_f of the flow network G, by using Depth First Search (DFS). Since the given flow f_0 is not a maximum flow for the flow network G. There must exist an augmenting path P in the residual graph G_f. Next, augment the flow f_0 by the augmenting path P which means that add the bottleneck value, the minimum edge flow capacity in the augmenting path P, for the forward edges in f_0 and subtract the bottleneck value for the backward edges in f_0. Finally, output the new flow f_1.

Algorithm:

```
given f_1(e) for any e

compute the residual graph G_f

find an augmenting path P in G_f

f_1 := augment f_0 using path P

output f_1
```

Proof of Correctness:

Lemma 3. *If f is a flow with no augmenting path P, then f is a maximum flow.*

Proof. Proof was given in the lecture. $\square$

Lemma 4. *If f is a flow whose residual graph G_f contains an augmenting path P with bottleneck capacity b, then augmenting path P yields a flow f whose value is $v(f) + b$. Especially, the value of this* *flow f is greater than the value of f, so f is not a maximum flow.*

Proof. Proof was given in the lecture. □

Running Time:

There are m edges and n vertices in the flow network G. The build of the residual graph takes $O(m+n)$ time. It takes $O(m)$ time to find the augmenting path P in the residual graph G_f by using DFS. In total, the time complexity is $O(m+n)$.