CS 4820, Spring 2018	**Homework 8, Problem 3**
Name: Rongguang Wang	
NetID: rw564	
Collaborators: Siyuao Liu (sl2928); Yihao Chen (yc2288)	

(3) *(12 = 2+10 points)* If G is a flow network and f is a flow in G, we say that f *saturates* an edge e if the flow value on that edge is equal to its capacity, i.e. $f(e) = c_e$. The Flow Saturation problem is the following decision problem: given a flow network G and a positive integer k, determine if there exists a flow f in G such that f saturates at least k edges of G. This problem asks you to prove that Flow Saturation is NP-complete.

(a) *(2 points)* What is wrong with the following incorrect solution?

> The Flow Saturation problem is in NP because it has a polynomial-time verifier that takes the pair (G, k) along with a flow f, and checks that f satisfies conservation and capacity constraints (in linear time). While checking capacity constraints it also keeps a counter of how many edges are saturated, and it reports "yes" if the conservation and capacity constraints are satisfied, and the number of saturated edges is at least k.
>
> To prove that Flow Saturation is NP-complete we reduce from Hamiltonian Path. Given a directed graph G_0 that is an instance of Hamiltonian Path, our reduction creates a new graph G consisting of a copy of G_0 together with four extra vertices $\{s, s', t', t\}$. G has two new edges (s, s') and (t', t), and it also has edges from s' to every vertex of G_0 and from every vertex of G_0 to t'. Finally we set all edge capacities to 1, and we treat this as an instance of Flow Saturation with source s, sink t, and parameter $k = n + 3$. The reduction takes linear time: if G_0 has n vertices and m edges, then G has $n + 4$ vertices and $m + 2n + 2$ edges, and it takes constant time to insert each vertex or edge into the adjacency list representation of G.
>
> To prove the correctness of the reduction, we show the following two statements.
>
> **If G_0 has a Hamiltonian path, then G has a flow that saturates $n + 3$ edges.** Indeed, if the Hamiltonian path P_0 in G_0 is from s_0 to t_0, then G contains a path P of length $n + 3$ from s to t, that begins with s, s', s_0, then traverses the entire path P_0 to reach t_0, then ends with t_0, t', t. Sending one unit of flow on this path P saturates all of its $n + 3$ edges.
>
> **If G has a flow that saturates $n + 3$ edges, then G_0 has a Hamiltonian path.** Indeed, since G has just a single unit-capacity edge leaving s, the max-flow value in G is equal to 1 and so a maximum flow consists of a single path from s to t. If this path saturates $n + 3$ edges, then two of its edges leave s and s', respectively, two of its edges come into t' and t, respectively, and the remaining $n - 1$ edges belong to G_0. Those $n - 1$ edges must form a Hamiltonian path in G_0.

Solution:

The proof reduces the Flow Saturation (FS) problem from Hamiltonian Path problem. The reduction direction is correct but the proof is wrong. It states that If G has a flow that saturates $n + 3$ edges, then, G_0 has a Hamiltonian Path. However, the flow produced may contain cycle which is definitely not a Hamiltonian Path. Therefore, the solution is incorrect.

(b) *(10 points)* Prove that the Flow Saturation problem is indeed NP-complete.

REMARK: Part (a) contains a valid proof that FLOW SATURATION **belongs to NP. Therefore, in doing part (b), you do not need to include that step in your solution.**

Solution:

The problem is aiming to proof that the Flow Saturation (FS) problem is NP-Complete. In FS problem, there is a flow network G, and a flow f in G. If the flow value $f(e)$ on one edge e is equal to its capacity c_e, then we call f saturate edge e. The problem is to determine whether there exists at least k edges that are saturated by flow f in graph G.

To prove that FS problem is NP-Complete, we reduce from Independent Set. Given an instance of Independent Set specified by a graph G and a parameter k, our reduction works as following. Let n denotes the number of vertices and let m denotes the number of edges in G. Equate the vertex set of G with the set $[n]$, and for $i = 1, \ldots, n$, let d_i denotes the degree of vertex i. Number the edges of G as $\{e_1, \ldots, e_m\}$. Our reduction constructs a flow network with a list of nodes and edges. A source node s and a sink node t. Edge nodes $e_1, \ldots, e_m$ which can be called layer 1. Vertex nodes $v_1, \ldots, v_n$ which can be called layer 2. Edges (s, w_i) of capacity 1 for $i = 1, \ldots, m$. Edges (w_i, v_j) of capacity 1 for each pair i, j such that i is an endpoint of edge i. Edges (v_j, t) of capacity 1 for $j = 1, \ldots, n$. The instance of FS problem constructed by our reduction consists of the network G and the parameter $k + 2m$. Our network has $n+m+2$ vertices and $n+3m$ edges. Constructing an edge with capacity c requires $O(\log c)$ time. Our graph has n edges of capacity at most n and $3m$ edges of capacity 1, so the running time of the reduction is $O(n \log n + m)$. In particular, the reduction runs in polynomial time. Let G' denotes the flow network produced by the reduction. To prove the correctness of the reduction, we need to show two things.

1. If G has an independent set of size k, then G' has a flow that saturates at least $k + 2m$ edges. Let s be an independent set of size k in G. For every edge e_j choose one endpoint $i(j)$ as follows: if e_j has an endpoint that belongs to S, then there must be only one such endpoint and we let $i(j)$ be that endpoint. Otherwise, select $i(j)$ to be an arbitrary endpoint of e_j. Now construct a flow f in G as follows. For every $j = 1, \ldots, m$, send one unit of flow from w_j to $v_{i(j)}$. Send zero flow on all other edges between layer 1 and layer 2. Send one unit of flow on every edge into t. Finally, for every $i = 1, \ldots, n$, the flow on edge (s, w_i) is set equal to the number of edges e_j such that $i(j) = i$. Note that this is also equal to the number of units of flow leaving v_i, and that it is bounded above by d_i. Hence, flow conservation is satisfied at v_i and the capacity constraint of edge (s, v_i) is satisfied. By construction flow f saturates the flowing $k + 2m$ edges: the edges (s, v_i) for all $i \in S$, the edges $(v_{i(j)}, w_j)$ and (w_j, t) for $j = 1, \ldots, m$. It is trivial to see that $(v_{i(j)}, w_j)$ and (w_j, t) are saturated, and to verify that (s, v_i) is saturated for i we observe that when i, every edge e_j incident to i chooses $i(j) = i$.

2. If G' has a flow that saturates at least $k + 2m$ edges, then G has an independent set of size k. For $j = 1, \ldots, m$, the only outgoing edge of w_j has capacity 1, and both of the incoming edges of w_j also have capacity 1. So, at most one of the incoming edges of w_j can be saturated by a flow. This means that every flow saturates at most $2m$ of the edges incident to layer 2. So if a flow succeeds in saturating $k + 2m$ edges, it must saturate at least k of the edges from s to layer 1. Let S denotes the set of all i such that edge (s, v_i) is saturated. We have argued that $|S| \leq k$, now show that S is an independent set in G. If e_j is any edge of G, let i, i' denote the endpoints of edge e_j. We must prove at least one of i and i' does not belongs to S. As noted earlier, at most one of the incoming edges w_j can be saturated. Assume without loss of generality that edge (v_i, w_j) is not saturated. Then, not all of the outgoing edges of v_i are saturated, so the combined outgoing flow of v_i is strictly less than d_i. By flow conservation, the incoming flow to v_i must also be strictly less than d_i, implying $v_i \notin S$ as claimed.