CS 4820, Spring 2018	**Homework 7, Problem 2**

Name: Rongguang Wang
NetID: rw564
Collaborators: Siyuao Liu (sl2928); Yihao Chen (yc2288)

(2) *(10 points)* At Ford-Fulkerson University there are many committees that need to be staffed with professors. The university has n professors organized into d departments; each professor belongs to only one department. There are m committees, and the following constraints must be satisfied when staffing the committees.

1. The required number of professors on committee k is specified by a positive integer r_k.
2. No professor is allowed to serve on more than c committees.
3. No committee is allowed to have more than one professor from the same department.
4. For each professor j, there is a list L_j of the committees on which he or she is qualified to serve. Professor j is not allowed to serve on committee k unless $k \in L_j$.

Design a polynomial-time algorithm to determine whether it is possible to staff each committee without violating any of the constraints listed above. If it is possible to staff the committees, your algorithm should output an assignment of professors to committees that satisfies all of the constraints. The input to the problem is specified by the numbers $n, d, m, r_1, \ldots, r_m$, and the lists $L_1, \ldots, L_n$.

Solution:

The problem is aiming to find an algorithm which can staffing the professors to commitees with several constraints. Here are the requirements from the problem. Firstly, there should be r_k professors on committee k. Secondly, any professor can serve at most c committees. Thirdly, any committee can be serve by at most one professor from the same department. Finally, for each professor j there is a list L_j of the committees on which he or she is qualified to serve, which means that professor j is not allowed to serve on committee k unless $k \in L_j$. Additionally, there are n professors, d departments and m committees in the problem. The problem can be solved by reducing into Ford-Fulkerson Algorithm. We need to construct a flow network G for the input of the algorithm. Firstly, a source node s and a sink node t should be created. Then, a set of professor nodes $\{u_j |\ j = 1, \ldots, n\}$, a set of intermediate nodes $\{v_{ki} |\ k = 1, \ldots,\ m\ \&\ i = 1, \ldots, d\}$, and a set of committee nodes $\{w_k |\ k = 1, \ldots, m\}$ should be created. Next, edges between the nodes should be connected. The edges between the source node s and the professor nodes u_j should be connected with capacity c. This set of edges enforces the constraint 2 that any professor can serve at most c committees. The edges between the professor nodes u_j and the intermediate nodes v_{ki} should be connected with capacity 1 for all k and i such that professor j belongs to department i and k is the index of the committees which belongs to the list L_j. The intermediate nodes v_{ki} are abstraction of professor and department which takes in professors and outputs department for committee nodes w_k. This set of edges enforces the constraint 4 that professor j is not allowed to serve on committee k unless $k \in L_j$. The edges between intermediate nodes v_{ki} and the committee nodes w_k should be connected with capacity 1 for all k and i. This set of edges (u_j, v_{ki}) enforces that the constraint 3 that any committee can be serve by at most one professor from the same department. Finally, the edges between the committee nodes w_k and sink node t should be connected with capacity r_k. This set of edges enforces the constraint 1 that there should be r_k professors on committee k. Now, the flow network graph G was built and we can run the Ford-Fulkerson Algorithm to find a maximum flow f. It is noticeable that all flow value on the edges of the graph G should be positive integers. Otherwise, the algorithm may not terminate. To check whether the maximum flow f meet the requirement of the problem, we should focusing on if the number of professors on committee k is r_k. If there exists an edge (w_k, t) such that $f(w_k, t) < r_k$ for any k, then it is impossible to staff the committees. Otherwise, the committees can be staffed by assigning professor j to committee k for every edge (u_j, v_{ki}) such that $f(u_j, v_{ki}) = 1$.

Algorithm:

```
given n professors, d departments, m committees, number of professors r_k
should be staffed on committee k and list of committees L_j that professor j
can serve on

build flow network graph G:

create nodes:

  a source node s and a sink node t
  n professor nodes u_j
  m*d intermediate nodes v_ki
  m committee nodes w_k

create edges:

  edges (s, u_j) pointing from source node to each professor nodes with capacity c

  edges (u_j, v_ki) pointing from each professor nodes to the intermediate nodes
  with department i professor j belongs to and committees in list L_j professor j
  can serve on with capacity 1

  edges (v_ki, w_k) pointing from each intermediate nodes to the committee node
  who has the same committee index k with capacity 1

  edges (w_k, t) pointing from each committee nodes to the sink node with capacity r_k


run Ford-Fulkerson Algorithm to find a maximum flow f

if there exists an edge (w_k, t) such that f(w_k,t) < r_k for any k,

  output null

else

  assign professor j to committee k for every edge (u_j,v_ki) such that f(u_j,v_ki)=1
```

Proof of Correctness:

Lemma 1. *For every valid committee assignment, there exists a flow f with value $\sum_k r_k$, and that for every integer-valued flow with at least this value in the flow network G.*

Proof. In a maximum flow f, $f(u_j, v_{ki}) = 1$ means that professor j in department i serves on committee k. $f(s, u_j) = c$ means professor j serves in c committees. $f(v_{ki}, w_k) = 1$ means some professors from department i servers on committee k. $f(w_k, t) = r_k$ means the number of professor serving on committee k is r_k. All other edges have flow value zero. Then, this flow f satisfies all flow conservation and capacity constraints. Its value equals to the net amount of flow going into t, which is equals to $\sum_k r_k$.□

Lemma 2. *The output of Ford-Fulkerson Algorithm can be transformed into a valid output of the problem.*

Proof. Considering the last step of the algorithm, assigning professor j in department i to committee k if and only if $f(u_j, v_{ki}) = 1$. According to the construction of the flow network, professor j can be assigned to committee k only if $k \in L_j$. Then, the flow on edges (v_{ki}, w_k) is equals to the number of professors from department i assigned to committee k. Since the capacity of these edges is 1, at most one professor from department i is assigned to committee k. Additionally, the flow on edges (w_k, t) is equals to the number of professors assigned to committee k with capacity r_k. Finally, the flow on edges (s, u_j) is equals to the number of committees that professor j is assigned to. Since the capacity of these edges is c, no professor is allowed to be assigned to more than c committees. □

Running Time:

There are $m \times d + n + m + 2 = m \cdot (d + 1) + n + 2$ which is $O(md + n)$ nodes and $n + mn + m + m$ which is $O(mn)$ edges in the flow network graph G. The construction of the network takes $O(1)$ for each node and edge. Finding the maximum flow using Ford-Fulkerson Algorithm takes $O(mV)$ where m is the number of edges and V is the maximum flow value. In this problem, the maximum flow value is bounded by mn since there are at most mn edges from the professor nodes layer to the intermediate nodes layer and each edge has capacity 1. So, the cut which has $\{s, u_1, \ldots, u_n\}$ on one sie and all other nodes on the opposite side has capacity mn. Substituting this upper bound for the max-flow value into the Ford-Fulkerson Algorithm running time bound, and recalling that the number of edges is $O(mn)$, we find that the running time is bounded by $O(m^2 n^2)$.