1. (20 pts) Given a directed graph $G = (V, E)$ with capacity $c(u, v) > 0$ for each edge $(u, v) \in E$ and demand $r(v)$ at each vertex $v \in V$, a routing of flow is a function f such that

   - for all $(u, v) \in E$, $0 \leq f(u, v) \leq c(u, v)$, and
   - for all $v \in V$,
   $$\sum_{u:(u,v)\in E} f(u, v) - \sum_{u:(v,u)\in E} f(v, u) = r(v),$$
   i.e., the total incoming flow minus the total outgoing flow at vertex v is equal to $r(v)$. Notice that the demand $r(v)$ can take positive value, negative value, or zero.

   (a) Show how to find a routing or determine that one does not exist by reducing to a maximum flow problem.

   (b) Suppose that additionally there is a lower bound $l(u, v) > 0$ at each edge (u, v), and we are looking for a routing f satisfying $f(u, v) \geq l(u, v)$ for all $(u, v) \in E$. Show how to find such a routing or determine that one does not exist by reducing to a maximum flow problem.

2. (20 pts extra credit) Even though in this class we focus on those greedy algorithms that generate optimal solutions, in general a greedy algorithm may not give an optimal solution. So, we are interested in those greedy algorithms that generate a good enough solution, i.e., not too far from the optimal solution. Let us consider one such problem as follows.

   Given subsets $S_1, S_2, \cdots, S_n$ of a set S of points and an integer m, a maximum $m-$cover is a collection of m of the subsets that covers the maximum number of points of S. Finding a maximum $m-$cover is a computationally hard problem. Give a greedy algorithm that achieves approximation ratio $1-1/e$; i.e., let y be the maximum number of points that can be covered by m subsets and x be the number of points that are covered by the m subsets generated by your algorithm, then give a greedy algorithm such that $x \geq (1 - 1/e)y$.

   Some useful hints:

   - You may want to use the inequality $(1 - 1/m)^m \leq 1/e$ for integer $m \geq 1$.
   - You may want to use induction at some point.