1. Suppose that each source $s_i$ in a multisource, multisink problem produces exactly $p_i$ units of flow, so that $f(s_i, V) = p_i$. Suppose that each sink $t_j$ consumes exactly $q_j$ units so that $f(V, t_j) = q_j$, where. Show how to convert the problem of finding a flow $f$ that obeys these additional constraints into the problem of finding a maximum flow in a single-source, single-sink flow network.

2. Given a flow network $G = (V, E)$, let $f_1$ and $f_2$ be functions from $V \times V$ to $\mathbb{R}$. The flow sum $f_1 + f_2$ is the function from $V \times V$ to $\mathbb{R}$ defined by $(f_1 + f_2)(u, v) = f_1(u, v) + f_2(u, v)$ for all $u, v \in V$. If $f_1$ and $f_2$ are flows in $G$, which of the three flow properties must the flow $f_1 + f_2$ satisfy, and which might it violate?

3. The edge connectivity of an undirected graph is the minimum number $k$ of edges that must be removed to disconnect the graph. For example, the edge connectivity of a tree is 1, and the edge connectivity of a cyclic chain of vertices is 2. Show that how the edge connectivity of an undirected graph $G = (V, E)$ can be determined by running a maximum-flow algorithm on at most $|V|$ flow networks, each having $O(|V|)$ vertices and $O(|E|)$ edges.

4. Let $P$ be a simple (not necessarily convex) polygon enclosed in a given rectangle $R$, and $q$ be an arbitrary point inside $R$. Design an efficient algorithm to find a line segment connecting $q$ to any point outside $R$ that this line intersects is minimum.

5. Let $P$ be a set of $n$ points in a plane. We define the depth of a point $p$ in $P$ as the number of convex hulls that need to be ‘peeled’ (removed) for $p$ to become a vertex of the convex hull. Design an $O(n^2)$ algorithm to find the depths of all points in $P$.

6. Given a set of $n$ points in the plane $P$. A straight forward or brute force algorithm will take $O(n^2)$ to compute a pair of closest points. Give an $O(n \log n)$ algorithm find a pair of closest points. You get a bonus if you can give an $O(n \log n)$ algorithm.

7. Extend Rabin-Karp method to the problem of searching a text string for an occurrence of any one of a given set of $k$ patterns? Start by assuming that all $k$ patterns have the same length. Then generalize your solution to allow the patterns to have different lengths.

8. Let $P$ be set of $n$ points in the plane. We define the depth of a point in $P$ as the number of convex hulls that need to be peeled (removed) for $p$ to become a vertex of the convex hull. Design an $O(n^2)$ algorithm to find the depths of all points in $P$.

9. The input is two strings of characters $A = a_1, a_2, ..., a_n$ and $B = b_1, b_2, ..., b_n$. Design an $O(n)$ time algorithm to determine whether $B$ is a cyclic shift of $A$. In other words, the algorithm should determine whether there exists an index $k$, $1 \leq k \leq n$ such that $a_i = b(k+i) \mod n$, for all $i$, $1 \leq i \leq n$. 
