Flows

1. A cut in an undirected graph is a partition of the vertex set, V, into two sets U, W. The capacity of the cut U, W is the sum of the capacities of the edges with one end-point in U and the other in W. How will you find the cut of minimum capacity?
2. You are given a directed graph $G = (V, E)$, source and sink $s, t \in V$and edge capacities $c: E \rightarrow \Re^{+}$. The goal is to delete an edge so as to reduce the maximum s-t flow in G as much as possible. In other words, find an edge $e \in E$ so that the maximum flow in $G = (V, E\backslash\{e\})$ is as small as possible. Give an algorithm to solve this problem.
3. Is the following statement true or false. Justify your answer. "For any flow network G and any maximum flow on G there is always an edge e such that increasing the capacity of e increases the maximum flow of the network."
4. Let $G = (V, E)$ be a flow network with source s, sink t and integer capacities. Suppose that we are given a maximum flow in G.
    a. Suppose that the capacity of a single edge $e \in E$ is increased by 1. Give an $O(V + E)$-time algorithm to update the maximum flow.
    b. Suppose that the capacity of a single edge $e \in E$ is decreased by 1. Give an $O(V + E)$-time algorithm to update the maximum flow.
5. Recall the maximum flow algorithm in which we routed flow along the shortest path in each step. A phase was defined as a sequence of steps in which the length of the shortest path remains the same. Show how to implement a phase in $O(n^2)$ time, where n is the number of vertices.
6. Consider a graph in which all edge capacities are 1. Show how to implement a phase in $O(m)$ time, where m is the number of edges.
7. Use the solution of the above problem to compute a maximum flow in a unit-capacity graph in $O(m\sqrt{m})$ time.
8. Use the solution to the above problem to compute a maximum matching in a bipartite graph on n vertices and m edges in $O(m\sqrt{n})$ time.