Problem 1 (20 Points). There are many common variations of the maximum flow problem. Here are four of them:

(a) There are many sources and many sinks, and we wish to maximize the total flow from all sources to all sinks.

(b) Each *vertex* has a capacity on the maximum flow that can enter it.

(c) Each edge has not only a capacity, but also a *lower bound* on the flow it must carry.

(d) The outgoing flow from each node v is not the same as the incoming flow, but is smaller by a factor of $(1 - \epsilon_v)$, where ϵ_v is a loss coefficient associated with node v. In this case, we wish to maximize the flow to the sink.

Each of these can be solved efficiently. Show this by reducing (a) and (b) to the original max-flow problem, and reducing (c) and (d) to linear programming.

Problem 2 (20 Points). A *vertex cover* of an undirected graph $G = (V, E)$ is a subset of the nodes which touches every edge — that is, a subset $S \subset V$ such that for each edge $(u, v) \in E$, one or both of u, v are in S.

Show that the problem of finding the minimum vertex cover in a *bipartite* graph reduces to maximum flow. *Hint: Can you relate this problem to the minimum cut in an appropriate flow network*

Problem 3 (20 Points). Suppose we have a bipartite matching problem with n boy and n girls. Hall's Theorem says that there is a perfect matching if and only if the following condition holds: any subset S of boys is connected to at least $|S|$ girls.

Prove this theorem. *Hint: the max-flow min-cut theorem should be helpful.*

Problem 4 (10 Points). Show that for any problem in NP, there is an algorithm which solves it in time $O(2^{p(n)})$, where n is the size of the input instance and $p(n)$ is a polynomial (which may depend on the problem).

Problem 5 (30 Points). In the NODE-DISJOINT PATHS problem, the input is an undirected graph in which some nodes have been specially marked: a certain number of "sources" $s_1, ..., s_k$ and an equal number of "destinations" $t_1, ..., t_k$. The goal is to find k node-disjoint paths (that is, paths which have no nodes in common) where the ith path goes from s_i to t_i. Show that that this problem is NP-Complete.

Here is a sequence of progressively stronger hints:

i. Reduce from 3SAT

ii. For a 3SAT formula with m clauses and n variables, use $k = m + n$ sources and destinations. Introduce one source/destination pair(s_x, t_x) for each variable x and one pair (s_c, t_c) for each clause c.

iii. For each 3SAT clause, introduce 6 new intermediate nodes, one for each literal occurring in that clause and one for its complement

iv. Notice that if the path from s_c to t_c goes through some intermediate node representing, say, an occurrence of variable x, then no other path can go through that node. What node would you like the other path to be forced to go through instead.

Total points: 100