1. Suppose a graph, in addition to capacities on edges, has a capacity on every vertex such that the total inflow (and therefore the total outflow from the node) cannot exceed the capacity of the vertex. The other constraints remain the same as in the standard maximum-flow problem. How will you find the maximum flow in this graph?

2. Suppose that the maximum flow in a graph G = (V, E) has already been computed. Then the capacity of only one edge increases by 1. Give an efficient algorithm to recompute the new flow in the graph.

3. Consider a $p \times q$ matrix X. Each element of X is a non-negative real number. However, the sum of the elements in any row and in any column is an integer (may be different for different rows and columns). Construct a $p \times q$ matrix Y such that each element of Y is a non-negative integer, and the sum of the elements in any row and in any column in Y is the same as the sum in the corresponding row and column respectively in X. You must use maximum flow to construct Y.

4. A college has N students X_1, X_2, ..., X_N , M departments D_1, D_2, ..., D_M, and P societies S_1, S_2, ..., S_P. Each student is enrolled in exactly one department, and is a member of at least one society. The college has a student association (like your gymkhana) with one member from each society (You can assume that every society has at least one member). However, the society members have to be chosen such that the student association has at most Q_k members from any department D_k. Design an algorithm to form the association. Will you be able to form such an association always? You must model the problem as a maximum flow problem.

5. Consider a large task T that can be decomposed into n subtasks T_1, T_2, ..., T_n. The subtasks are to be performed on two machines M_1 and M_2. A subtask can be done on any of the two machines, but will incur different costs on the two machines. Specifically, subtask T_i incurs a cost C_i if it is done on machine M_1, and cost D_i if it is done on machine M_2. Some pairs of these subtasks need to communicate among themselves. If such a pair of subtasks T_i and T_j are run on the same machine, the communication cost incurred is 0, otherwise, they incur a communication cost of P_{ij}. If a pair of subtasks do not need to communicate, they incur no communication cost irrespective of which machines they are run on. Find an assignment of the subtasks to the two machines so that the total cost of executing T is minimized.

6. Demonstrate by an example that the stable matching in a bipartite graph need not be unique.

7. Let G = (M, W, E) be an instance of the stable matching problem (along with the preference lists). For m in M, define

valid(m) = {w $\in$ W | (m,w) $\in$ some stable matching}

and

best(m) = the woman of earliest preference in valid(m).

A matching is called man-optimal if every m is paired with best(m). Prove that the Gale–Shapley algorithm produces the man-optimal matching.

8. Prove that every regular bipartite graph has a perfect matching.