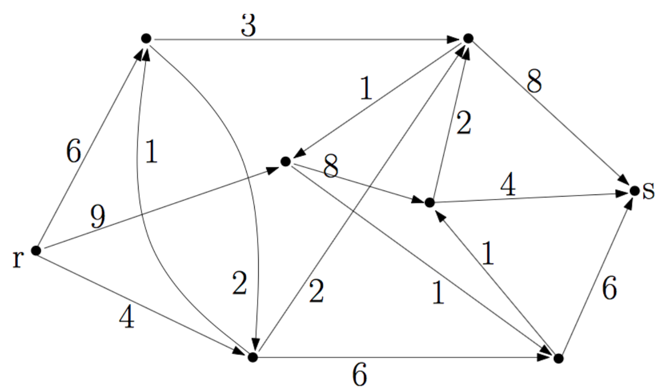
Advanced Algorithms

Problem set #1
Due date: 21 Aban 98 (via lms.iut.ac.ir)

Solutions to the problems on this set might be found on the Internet. It is not acceptable to copy such solutions. Write with your own notation and formulation.

Problem 1. Find the maximum flow and the minimum cut in the following graph:



Problem 2. Which of the following statements about flows are true and which are false? Justify your answer with a (short) proof or counterexample.

(a) If all directed edges in a network have distinct capacities, then there is a unique maximum flow.

(b) If we replace each directed edge in a network with two directed edges in opposite directions with the same capacity and connecting the same vertices, then the value of the maximum flow remains unchanged.

Problem 3. Translate maximum-flow instance in problem 1 to a linear programming formulation.

Problem 4. A factory manager has a table(following table) which shows how much profit is accomplished in an hour when each of six men operate each of six machines. Note that: man 6 cannot work on machine 1 because this task requires good eyesight; man 2 cannot work on machine 3 because he is allergic to dust; and man 5 cannot work on machine 6 because this job requires two hands. You are required to find the maximum profit assignment.

		Men					
		1	2	3	4	5	6
Machines	1	7	7	8	6	7	—
	2	8	5	8	6	5	5
	3	6	—	7	5	6	5
	4	5	4	5	5	4	4
	5	6	6	7	7	6	6
	6	7	8	7	6	—	6

Problem 5. 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* also 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 u is not the same as the incoming flow, but is smaller by a factor of $(1 - \epsilon_u)$, where ϵ_u is a loss coefficient associated with node u.

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.

[**Hint**: "reducing" for (a) and (b) simply means using Max-Flow as a subroutine to solve the problem at hand. A hint for (a) and (b) is that you may want to slightly alter the graph structure with additional nodes or edges that make the problem a traditional max-flow one.]

Problem 6. 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 how to determine the edge connectivity of an undirected graph $G = (V, E)$ by running a maximum-flow algorithm on at most $|V|$ flow networks, each having $O(|V|)$ vertices and $O(|E|)$ edges.

Problem 7. The *bisection* of a graph is defined as the smallest number of crossing edges when dividing the vertices of the graph into two sets of equal size (there is no connectivity requirement for the sets). Write an integer program that computes the bisection of a graph with an even number of vertices.

Problem 8. You are given an undirected graph $G = (V, E)$ with edge weights w_e for every $e \in E$. An $(s,t) - cut$ is a partition of the graph into two sets of vertices X and $V \setminus X$. The capacity of an $(s,t) - cut$ X is the sum of edge weights crossing that cut (that is, with exactly one endpoint in X). The minimum $(s,t) - cut$ is the cut with the minimum capacity.

1. Write an integer program to compute the minimum $(s,t) - cut$ in G.

2. Relax your program from part a to a linear program, and take it's dual. Interpret your result as an LP for the the maximium flow problem.

Problem 9. You are given a directed network $G = (V, E)$ with edge capacities c_e for every $e \in E$. Instead of a single (s, t) pair, you are given multiple pairs $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$, where the s_i are sources of G and the t_i are sinks of G. You are also given k demands $d_1, d_2, \ldots, d_k$. The goal is to find k flows $f^{(1)}, f^{(2)}, \ldots, f^{(k)}$ with the following properties:

1. Every $f^{(i)}$ is a valid flow from s_i to t_i.
2. For each edge e, the total flow across that edge is less than its capacity.
3. Each flow routes at least its demand, that is, $f^{(i)} \geq d_i$.
4. The total flow is maximized.

Can you formulate this problem as a simple max flow problem by transforming the graph? Can you formulate this problem as a linear program? Explain your answer (and give the formulation if you can).

Problem 10. A *line* in a matrix is a row or column of the matrix. Show that the minimum number of lines to cover all nonzero entries of a matrix (not necessarily square) is equal to the maximum number of nonzero entries, no two of which lie in a common line.

Good Luck.