COMS 4231: Analysis of Algorithms I, Fall 2018

Problem Set 6, due Sunday December 2, 11:59pm on Courseworks.

Note the due date: We plan to post solutions on Monday December 3, in preparation for the final, so *no late homeworks will be accepted.*

Please follow the homework submission guidelines posted on Courseworks.

- *As usual, for each of the algorithms that you give, include an explanation of how the algorithm works and show its correctness.*
- *Make your algorithms as efficient as you can, state their running time, and justify why they have the claimed running time. All time bounds below refer to worst-case complexity, unless specified otherwise.*

Problem 1. We are given a directed acyclic graph G, by its adjacency list representation, and two nodes s and t. Give an algorithm that computes the number of paths from s to t; you do not have to list explicitly the paths, just print the number. The algorithm should run in time $O(n+e)$, where n is the number of nodes and e the number of edges of the graph.
(Hint: If $d(u)$ denotes the number of paths from node u to node t, derive an equation that expresses $d(u)$ in terms of the quantities $d(v)$, $v \in \mathrm{Adj}(u)$.)

Problem 2. There is a set N of cities that a salesman may visit. If he visits city i he makes profit $p_i > 0$. The transportation cost from city i to city j is $c_{ij} > 0$. The salesman wants to find a cyclic route on a subset of cities such that the ratio of profit to cost is maximized. To this end, consider the directed graph $G=(N,E)$ whose nodes are the cities and which has an edge between every pair of cities. For any simple cycle $C = (N_C, E_C)$ in this graph, the profit-to-cost ratio is $r(C) = \dfrac{\sum_{j \in N_C} p_j}{\sum_{(i,j) \in E_C} c_{ij}}$.

Let r^* be the maximum ratio achievable by a cycle. One way to determine r^* is by binary search: by first guessing some ratio r and then testing whether it is too large or too small.
Consider any $r>0$. Let G_r be the weighted graph (G,w) where each edge (i,j) of G is given weight $w_{ij} = rc_{ij} - p_j$.
a. Show that if there is a cycle in G_r of negative weight then $r < r^*$.
b. Show that if all cycles of G_r have positive weight then $r > r^*$.
c. Give an efficient algorithm that takes as input the profits p_i, the costs c_{ij}, and a desired accuracy $\varepsilon>0$ and returns a cycle C for which $r(C) \geq r^* - \varepsilon$. Justify the correctness of your algorithm and analyze its running time in terms of $|N|$, ε, and $R = \max_{(i,j) \in E} (p_j / c_{ij})$.

Problem 3. a. If G=(N,E) is an undirected graph and u,v are two nodes, denote by $\mu(u,v)$ the maximum number of edge-disjoint paths from u to v (i.e. paths that do not share any edges). Show that $\mu(u,v)$ can be computed in polynomial time.

b. For an undirected graph G and nodes u,v, denote by $\lambda(u,v)$ the minimum number of edges whose removal disconnects u from v (i.e. there is a set L of $\lambda(u,v)$ edges such that the graph G'=(N,E-L) has no path from u to v). Prove that for every graph G and every pair of nodes u,v, $\mu(u,v)=\lambda(u,v)$.

c. The *edge connectivity* of a connected undirected graph G=(N,E), denoted $\lambda(G)$, is the minimum number of edges whose removal disconnects the graph. Observe that $\lambda(G)=\min\{\lambda(u,v) \mid u,v \in N\}$. Show that the edge connectivity can be computed by running a maximum flow algorithm on at most |N| flow networks, each having O(|N|) nodes and O(|E|) edges.

Problem 4. An *Integer Linear Program* (ILP) is the problem of optimizing (minimizing or maximizing) a linear function of a set of variables, subject to a set of linear inequality and equality constraints, with the additional restriction that the variables take only integer values. A 0-1 ILP is an ILP in which the variables are restricted to take value 0 or 1, i.e., the ILP includes (among others) constraints $x_i \geq 0$, $x_i \leq 1$, and the restriction "x_i integer" for all variables x_i.
a. Recall the Node Cover problem: Given an undirected graph G=(N,E), find a minimum-size subset S of nodes such that every edge contains a node in S.
Express the Node Cover problem as a 0-1 Integer Linear Program. In particular, given any undirected graph G, construct a 0-1 ILP that has a variable x_i for every node i of the graph G and has a suitable objective function and set of constraints such that a (0-1) vector x is an optimal solution to the 0-1 ILP if and only if the set of nodes $\{ i \mid x_i = 1\}$ is a minimum node cover of G.

b. Let *IP*(*G*) be your 0-1 ILP of part (a) for a graph *G,* and let *P*(*G*) be the Linear Program that is obtained from *IP*(*G*) by removing the integrality restriction on the variables, i.e. the LP that contains all the other constraints including the inequalities $x_i \geq 0$, $x_i \leq 1$. Is there a graph *G* such that *IP*(*G*) and *P*(*G*) have different optimal values? Either exhibit such a graph, or prove that *IP*(*G*) and *P*(*G*) have equal optimal values for all *G*.

c. Is there a bipartite graph *G* such that *IP*(*G*) and *P*(*G*) have different optimal values? Either exhibit such a graph, or prove that *IP*(*G*) and *P*(*G*) have equal optimal values for all bipartite graphs *G*.
(Hint: Compare the optimal value of *P*(*G*) to the size of the maximum matching.)

Exercises for your practice (do not turn in):
Practice the graph algorithms that we learned on some graphs of your choice. For example, do exercises 24.1-1, 24.2-1, 24.3-1, 25.2-1, 26.2-3, 26.2-4.