COMS 4231: Analysis of Algorithms I, Fall 2020

Problem Set 4, due Friday, October 16, 11:59pm on Gradescope.

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 [15 points] A weighted graph may have many different shortest (minimum-weight) paths between two nodes. In this case we may prefer among them one that has the minimum number of edges. Modify Dijkstra's algorithm to solve the following problem:
Input: A weighted directed graph G=(N,E) with positive weights w: $E \rightarrow R_+$ on its edges, and two nodes *s*, *t.*
Output: A minimum-weight path from *s* to *t* that contains as few edges as possible.

Problem 2 [25 points] We are given a set *V* of *n* variables $\{x_1, x_2, \ldots, x_n\}$ and a set *C* of *m* weak and strict inequalities between the variables, i.e., inequalities of the form $x_i \le x_j$ or $x_i < x_j$. The set *C* of inequalities is called *consistent* over the positive integers Z^+ iff there is an assignment of positive integer values to the variables that satisfies all the inequalities.
For example, the set $\{x_1 \le x_3,\ x_2 < x_1\}$ is consistent, whereas $\{x_1 \le x_3,\ x_2 < x_1,\ x_3 < x_2\}$ is not consistent.
a. Given a set *C* of *m* inequalities between *n* variables as above, let *G* be the directed graph with node set $V = \{x_1, x_2, \ldots, x_n\}$ and edge set $E = \{(x_i, x_j) \mid C \text{ contains } x_i \le x_j \text{ or } x_i < x_j\}$, i.e., *G* has one node for every variable and one edge for every inequality of *C*. Show that the set *C* is not consistent if and only if it contains a strict inequality $x_u < x_v$ such that the graph *G* has a path from x_v to x_u.
b. Give an O(*n*+*m*)-time algorithm to determine whether a given set *C* of inequalities is consistent over the positive integers.
c. If the set of inequalities has a solution over the positive integers, then it has a unique minimum solution, i.e. a solution in which every variable has the minimum value among all possible solutions. For example the minimum solution for the set $\{x_1 \le x_3,\ x_2 < x_1\}$ is $x_1=2$, $x_2=1$, $x_3=2$. Give an O(*n*+*m*)-time algorithm to compute the minimum solution of a given consistent set *C*.

Problem 3 [20 points] We are given a weighted graph G and a minimum spanning tree T of G. Both G and T are given by adjacency list representations.
a. Suppose that we increase the weight of an edge (x,y) of T. Give an $O(n+e)$-time algorithm that computes the minimum spanning tree of the modified weighted graph, where n is the number of nodes and e the number of edges of G.
b. Suppose that we decrease the weight of an edge (u,v) that is not in T. Give an $O(n)$-time algorithm that computes the minimum spanning tree of the modified weighted graph.

Problem 4 [15 points] 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 $C(u)$ denotes the number of paths from node u to node t, derive an equation that expresses $C(u)$ in terms of the quantities $C(v)$, $v \in \text{Adj}(u)$.)

Problem 5. [25 points] 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 greater or smaller than r^*.
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 $R = \max_{(i,j)\in E} (p_j / c_{ij})$.

Exercises for your practice (do not turn in):

Practice the graph algorithms that we learn on some graphs of your choice.
For example, draw an arbitrary directed or undirected graph, pick a source node and apply Breadth-First Search. Compute the BFS tree, the distances, and the partition of nodes into layers. Does the tree depend on the order in which nodes appear in the

adjacency lists? (Recall that the adjacency list of a node contains the adjacent nodes in arbitrary order.) How about the distances and the layers?

Similarly with Depth-First Search. Compute the DFS tree, the discovery and finish times of the nodes, classify the edges (tree, forward, back, cross). Compute the strongly connected components for a directed graph. For example, do exercises 22.3-2 and 22.5-2.

Draw a weighted undirected graph, and apply Prim's and Kruskal's algorithm to it.

Similarly, practice Dijkstra's algorithm, Bellman-Ford and all the other graph algorithms that we did in class. For example, do exercises 24.1-1, 24.2-1, 24.3-1, 25.2-1, 26.2-3, 26.2-4.