B505/I500: Applied Algorithms

HW4 (Due: **Apr. 8 Monday 5pm**)

https://iu.instructure.com/courses/1771436

(The HW contains 20 bonus points)

1. (15 pts) Devise a linear time algorithm to solve the following problem: given a directed acyclic graph G, check if G has a directed path that visits every vertex once and only once.

2. (15 pts) A feedback edge set of an undirected graph G=(V, E) is a subset of edges E'$\subseteq$E that intersects every cycle of the graph. Thus, removing the edges in E' will render the graph acyclic. Give an efficient algorithm for the following problem:
   Input: Undirected graph G=(V, E) with positive edge weights w_e
   Output: A feedback edge set E'$\subseteq$E of minimum total weight $\Sigma_{e \in E'} w_e$

3. (15 pts) A bipartite graph is a graph G=(V, E) whose vertices can be partitioned into two non-overlapping sets ($V = V_1 \cup V_2$ and $V_1 \cap V_2 = \emptyset$) such that there are no edges between vertices in the same set (e.g., if u, v $\notin V_1$, then there is no edge between u and v). Devise a linear-time algorithm to determine whether an undirected graph is bipartite.

4. (20 pts) Suppose a CS curriculum consists of n courses, all of which are mandatory. The prerequisite graph G has a node for each course, and an edge from course v to course w if and only if v is a prerequisite for w. Devise an algorithm that works directly on this graph representation to compute the minimum number of semester necessary to complete the curriculum (assuming that a student can take any number of courses in a semester). The running time of your algorithm should be linear.

5. (15 pts) Professor Luke suggests the following algorithm for finding the shortest path from node s to node t in a directed graph with some negative edges: add a large constant to each edge weight so that all the weights become positive, then run Dijkstra's algorithm starting at node s, and return the shortest path found to node t. Is this a valid algorithm? If yes, prove it is correct. Otherwise, give a counterexample.

6. (20 pts) Devise an algorithm that takes as input a direct graph with positive edge lengths, and returns the length of the shortest cycle in the graph (if the graph is acyclic, it should say "no cycle". Your algorithm should take time at most $O(|V|^3)$.

7. (20 pts) You are given a set of cities, along with the pattern of highways between them, in the form of an undirected graph G=(V, E). Each stretch of highway e$\in$E connects two of the cities, and you know its length in miles, L_e. You want to get from city s to city t. There is one problem: your car can only hold enough gas to

cover L miles. There are gas stations in each city, but not between cities. Therefore, you can only take a route if every one of its edges has length $L_e \leq L$. a) Given the limitation on your car's fuel tank capacity, show how to determine in linear time whether there is a feasible route from s to t. b) You are now planning to buy a new car, and you want to know the minimum fuel tank capacity (in terms of the miles in coverage without re-fueling) that is needed to travel from s to t. Devise an $O((|V| + |E|) \log |V|)$ algorithm to determine this. (Hint: you may modify Dijkstra's algorithm).