CS 430 – FALL 2017
INTRODUCTION TO ALGORITHMS
HOMEWORK #7
DUE Fri Dec 1, 11:25am

1. (3 points) You are helping a group of ethnographers analyze some oral history data they have collected by interviewing members of a village to learn about the lives of people who lived there over the last two hundred years. From the interviews, you have learned about a set of people, all now deceased, whom we will denote P1, P2, P3,…, Pn. The ethnographers have collected several facts about the lifespans of these people:
   a) Pi died before Pj was born
   b) Pi and Pj were both alive at the same time
However, the ethnographers are not sure that there facts are correct. So they'd like you to determine whether the data they have collected is at least internally consistent, in the sense that there could have existed a set of people for which all the facts simultaneously hold. Describe an algorithm to answer the ethnographers' problem. Your algorithm should output whether all the facts are consistent, or output the subset of facts that may be inconsistent. Hint: construct a bipartite directed graph in which there are 2n nodes representing, respectively, dates of birth and death of the n people. Draw edges from one node to another when we have a fact that states that the former precedes the latter. Clearly a person's birth precedes his/her death, so that would be an edge. What does it mean if two people are claimed to have been alive at the same time?

2. (3 points) The **clustering coefficient** $C(G)$ of a simple graph G is the probability that if u and v are neighbors and v and w are neighbors, then u and w are neighbors, where u, v, and w are distinct vertices of G.
7a. We say that three vertices u, v, and w of a simple graph G form a triangle if there are edges connecting all three pairs of these vertices. Find a formula for $C(G)$ in terms of the number of triangles in G and the number of paths of length two in the graph. [*Hint:* Count each triangle in the graph once for each order of three vertices that form it.]
7b. Explain what the clustering coefficient measures in each of these graphs.
- the Hollywood graph (used for the six degrees of Kevin Bacon problem)
- the graph of Facebook friends
- the graph representing the routers and communications links that make up the worldwide Internet

3. (2 points) 3a. Show how on a directed graph that depth first search starting at vertex u can result in vertex v not being reachable from u even though both u and v have both incoming and outgoing edges.
3b. If a directed graph contains a path from u to v, show that it is not necessary that the s(v) < f(u). s() is the depth first search start time and f() is the depth first search finish time

4. (3 points) Bob loves foreign languages and wants to plan his course schedule to take the following nine language courses: LA15, LA16, LA22, LA31, LA32, LA126, LA127, LA141, and LA169. The course prerequisites are:
LA15: (none)   LA16: LAl5 is prerequisite
LA22: (none)   LA31: LAl5 is prerequisite
LA32: LA16 and LA31 are prerequisites   LA126: LA22 and LA32 are prerequisites
LA127: LAl6 is prerequisite   LA141: LA22 and LAl6 are prerequisites
LA169: LA32 is prerequisite
Find a sequence of courses that allows Bob to satisfy all the prerequisites.

5. (3 points) Prim's and Kruskal's algorithms both "grow" a minimum spanning tree of a graph by selecting edges to add to the tree in a specified, greedy order. Design an efficient algorithm to "prune" a graph and yield a minimum spanning tree by removing edges from the graph in a specified, greedy order. Prove optimal substructure and the Greedy Choice Property.

6. (3 points) Shortest Path Verification - Your roommate has written a program to implement Dijkstra's shortest path algorithm. Design and analyze a linear time algorithm to check your roommate's algorithm's results. That is, given a graph G = (V,E), a source vertex s, and your roommate's values of v.d and v.pi for every vertex v ∈ V , your algorithm must verify their correctness or find a value that is wrong.

7. (3 points) Exercise 24.1-1 (only first question, NOT "Now, change the weight of edge (z, x) to 4 and run the algorithm again, using s as the source.") Run the Bellman-Ford algorithm on the directed graph of Figure 24.4, using vertex z as the source. In each pass, relax edges in the same order as in the figure, and show the d and π values after each pass.

8. (5 points) Compute D(1), D(2), D(3), D(4) and D(5) using Floyd-Warshall algorithm to compute minimum weight paths between all vertices in the below graph. Also keep track of the predecessor vertices (last vertex in the shortest paths).