CSCE 629-601 Analysis of Algorithms

Fall 2022

Instructor: Dr. Jianer Chen
Office: PETR 428
Phone: (979) 845-4259
Email: chen@cse.tamu.edu
Office Hours: MWF 3:50pm–5:00pm

Teaching Assistant: Vaibhav Bajaj
Office: EABC 107B
Phone: (979) 739-2707
Email: vaibhavbajaj@tamu.edu
Office Hours: T; 2pm-3pm, TR: 4pm-5pm

Assignment # 3
(Due October 14)

Remark. In the following questions, you can assume that your graphs are connected.

1. Another way to perform topological sorting on a directed acyclic graph $G = (V, E)$ is to repeatedly find a vertex of in-degree 0, output it, and remove it and all of its outgoing edges from the graph. Develop an $O(|V| + |E|)$-time algorithm using this approach. Your algorithm should also be able to tell when the input graph has cycles.

2. Let G be a directed graph with strongly connected components C_1, C_2, …, C_k. The *component graph* G^c for G is a directed graph of k vertices w_1, w_2, …, w_k such that there is an edge from w_i to w_j in G^c if and only if there is an edge from some vertex in C_i to some vertex in C_j. Develop an $O(|V|+|E|)$-time algorithm that on a given directed graph $G = (V, E)$ produces the component graph G^c for G. Make sure that there is at most one edge between two vertices in the component graph G^c.

3. Design algorithms for $\mathrm{Max}(H)$, $\mathrm{Insert}(H, a)$, and $\mathrm{Delete}(H, i)$, where the set H is stored in a max-heap, a is the element to be inserted into the heap H, and i is the index of the element in the heap H to be deleted. Analyze the complexity of your algorithms.

4. Consider an extended version of the Shortest-Path problem. Suppose that you want to traverse from city s to city t. In addition, for some reason, you also need to pass through cities x, y, and z (in any order) during your trip. Your objective is to minimize the cost of the trip. The problem can be formulated as a graph problem as follows: Given a positively weighted directed graph G and five vertices s, t, x, y, z, find a path from s to t that contains the vertices x, y, z such that the path is the shortest over all paths from s to t that contain x, y, z, assuming that these paths are allowed to contain repeated vertices and edges. Develop an $O(m \log n)$-time algorithm for this problem. (**Hint.** In this question, you can assume Dijkstra's Shortest-Path algorithm.)