CSCE-629 Analysis of Algorithms

Spring 2019

Instructor: Dr. Jianer Chen
Office: HRBB 315C
Phone: 845-4259
Email: `chen@cse.tamu.edu`
Office Hours: T,Th 11:00 am–12:30 pm

Teaching Assistant: Qin Huang
Office: HRBB 315A
Phone: 402-6216
Email: `huangqin@email.tamu.edu`
Office Hours: MWF 3:00 pm–4:00 pm

Assignment # 2
(Due February 26, 2019)

1. Design algorithms for Min(H), Insert(H, a), and Delete(H, i), where the set H is stored in a 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.

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

2. Write the psuedo-code for the Dijstra's algorithm that solves the SINGLE-SOURCE SHORTEST PATH problem. Analyze the complexity of the algorithm (you can assume that the algorithm uses a heap for fringes and you can use your results in Question 1 directly). Give a formal proof that the algorithm works correctly when the edge weights are all non-negative.

3. Develop a linear-time (i.e., $O(m)$-time) algorithm that solves the SINGLE-SOURCE SHORTEST PATH problem for graphs whose edge weights are positive integers bounded by 10. (**Hint.** You can either modify Dijstra's algorithm or consider using Breath-First-Search.)

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.