

SSN College of Engineering

Department of Computer Science and Engineering

CS1403 — Design and Analysis of Algorithms

2019 – 2020

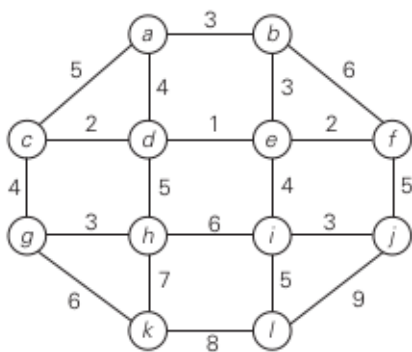
Session — 07

February 5, 2020

-
- This homework is due by 4pm on February 05, 2020
 - Grace period may be given up to midnight of February 06, 2020
 - You can upload only one ZIP file
 - The naming convention is “<Your first name (first letter capital and all the other letters small)>-CS1403-S07.zip”
 - The questions marked as “OPTIONAL” are, as the name implies, optional! Complete your core assignment first and attempt the optional problem only if you have sufficient time.
-

1. Given a weighted graph $G = (V, E)$, and a distinguished vertex $s \in V$ (source vertex), find the shortest weighted path from s from every other vertex in G .

An instance of this problem is shown below:



Consider node a as the source vertex and find the shortest paths from node a to all the other nodes.

- (a) Implement Dijkstra's algorithm with time complexity not exceeding $O(|E| \log |V|)$. You should use your minheap implementation from the previous session to achieve this time complexity. Note that you need to maintain references to every node in the heap to effectively implement the decreaseKey operation.

- (b) Let us assume that you have failed to implement `decreaseKey` and `increaseKey` operations on the minheap. How will you rewrite your code for Dijkstra's algorithm to use such a minheap implementation that has only `buildHeap`, `insert`, and `deleteMin` operations. Note that your time complexity should still not exceed $O(|E| \log |V|)$ (but you may use more space!)
- (c) (OPTIONAL) Suppose that the input graph is a Directed Acyclic Graph (DAG). Can the single-source shortest paths problem be solved in linear time?
- (d) (OPTIONAL) Suppose there is more than one minimum path from vertex s to some vertex t . Then the path with the minimum number of edges must be chosen. How will you modify your implementation to meet this requirement?
- (e) (OPTIONAL) Read about the Fibonacci heaps and their implementations. Explore if Fibonacci heaps may be used to provide a better implementation for Dijkstra's algorithm.