1. Discuss any of these problems (or any of the graph algorithms we’ve covered so far).

(a) Consider the following weighted, directed graph:

i. Is this a DAG (directed acyclic graph)?

ii. If you run a BFS starting from vertex C in what order do you visit the vertices? Assume adjacent vertices are returned in alphabetical order, i.e., would be added to the queue in alphabetical order.

iii. If you run Dijkstra’s algorithm with source vertex s, what is the order in which vertices are removed from the priority queue? Break ties alphabetically.

(b) If we modify Bellman-Ford so that it updates \(d[v]\) if \(d[v] \geq d[u] + w(u, v)\) (instead of if \(d[v] > d[u] + w(u,v)\)), does it still produce correct shortest-path weights? Explain.
(c) Assume you (1) have a weighted, directed graph $G$ where all the weights are integers between 1 and $W$ and (2) have an implementation of a min-priority queue that supports extract-min and decrease-key in time $O(\log \log k)$ assuming all the values are integers between 1 and $k$.

i. What is the running time of Prim’s on this graph with this data structure? Explain.

ii. What is the running time of Kruskal’s on this graph if you use a linear-time sort based on the additional information about the weights? Explain.

2. We’re almost 2/3rds of the way through the course.

   (a) What has been the most challenging thing about this course?

   (b) Are there any topics that you hope we’ll cover in the remaining part of the course?

3. Was everyone in the group at the meeting and, if not, who was missing?