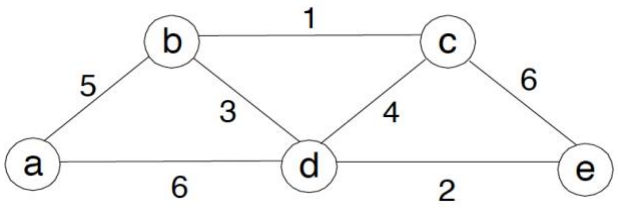
CS 477/677 Analysis of Algorithms

Spring 2024

Due May 7, 2024

1. (U&G-required) [20 points]

Answer the questions below regarding the following graph:

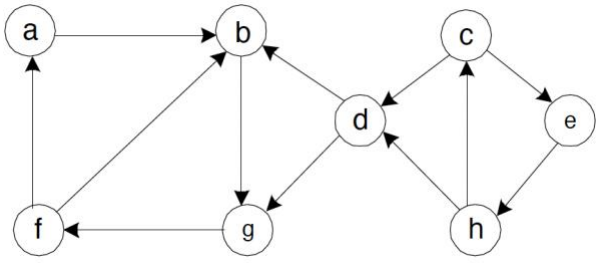


a) [10 points] In what order are edges added to the Minimum Spanning Tree (MST) using Kruskal's Algorithm? List the edges by giving their endpoints.

b) [10 points] In what order are edges added to the MST using Prim's Algorithm starting from vertex **e**? List the edges by giving their endpoints.

2. (U&G-required) [20 points]

Find the strongly connected components of the graph below, using the algorithm discussed in class. In the main loop for the first depth first search, consider the vertices in alphabetical order.



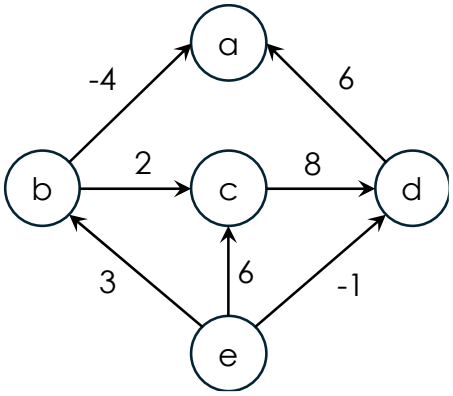
3. (U&G-required) [30 points]

a) [10 points] Explain how we could find that a graph has a cycle using breadth first search.

b) [20 points] For the two graph traversal algorithms breadth first search and depth first search, does either one **always** find a cycle faster than the other? If **yes**, indicate which one is better and give a justification. If **no**, give two examples in support of your answer.

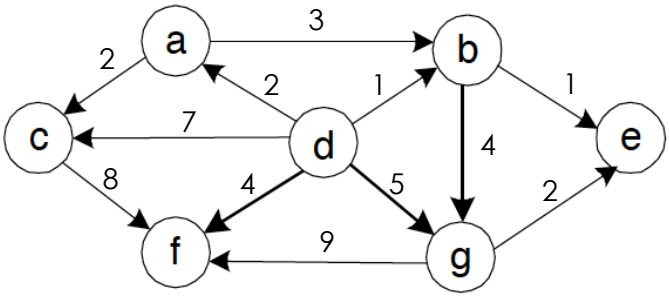
4. (U&G-required) [30 points]

a) [10 points] Run the topological sort algorithm on the graph below and indicate the final ordering of vertices obtained. In the main DFS loop, choose white vertices in alphabetical order (thus, the initial source will be vertex a). Also, assume that vertices are given in alphabetical order in the adjacency list.



b) [10 points] Run DAG-SHORTEST-PATHS on the directed graph **from part (a)**, using vertex *e* as the source.

c) [10 points] Show how Dijkstra's algorithm runs on the graph below, with vertex *d* as the source.



5. (G-required) [20 points]

a) [10 points] Indicate what is the running time efficiency of the depth first search algorithm if the graph is stored using the adjacency matrix representation.

b) [10 points] Indicate what is the running time efficiency of the depth first search algorithm for both the adjacency list and adjacency matrix representation if the graph is sparse: $|E| \in O(|V|)$, where E is the number of edges and V the number of vertices of the graph.

Extra credit:

6. [20 points] Explain what adjustments would have to be made (if any) to Djikstra's algorithm in order to find the shortest path between two vertices (the **single-pair shortest path problem**) in a weighted graph (directed or undirected). Assume that the graph has positive edge weights.