CSE 417 Midterm, Wednesday, February 8, 2023

NAME: ____________________________ UW Net ID: ____________________________

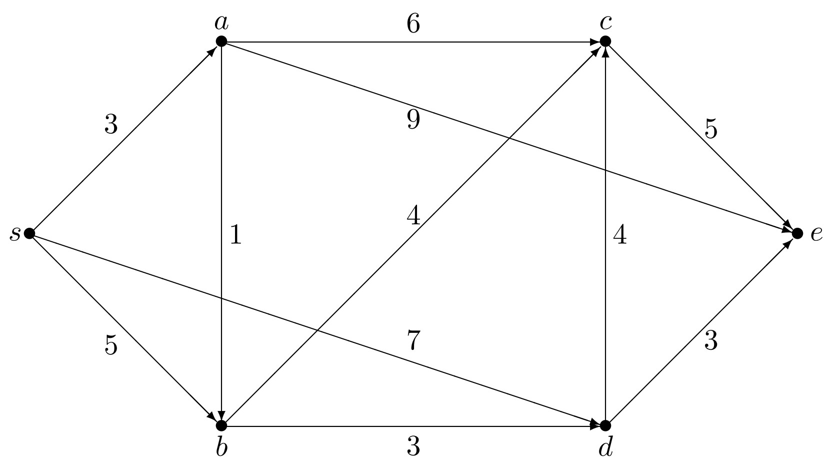
Instructions:

- Closed book, closed notes, no calculators
- Time limit: 50 minutes
- Answer the problems on the exam paper.
- If you need extra space use the back of a page
- Problems are not of equal difficulty, if you get stuck on a problem, move on.
- "Justify your answer" means give a short and convincing explanation. Depending on the situation, justifications can involve counter examples, or cite results established in the text or in lecture.

1	/10
2	/10
3	/10
4	/10
5	/10
6	/10
Total	/60

Problem 1. Dijkstra's Algorithm (10 points):

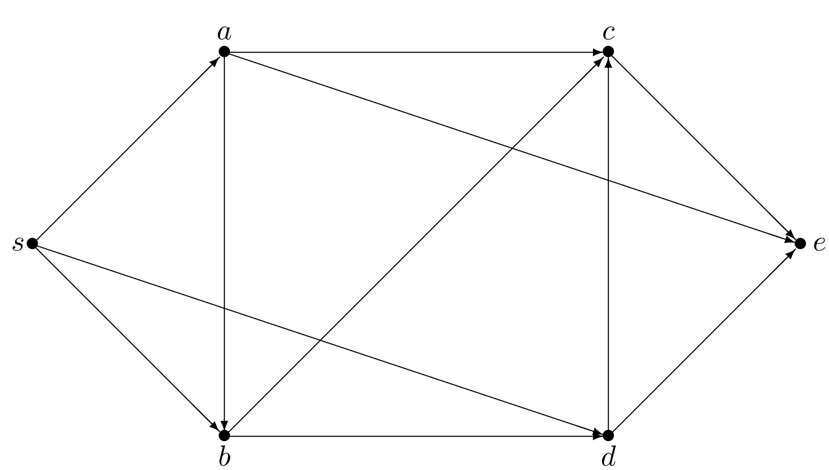
Use the following graph to simulate Dijkstra's algorithm starting from the vertex s.



a) Simulate Dijkstra's shortest path algorithm on the graph above by filling in the table. Row 0 should give the initialization values.

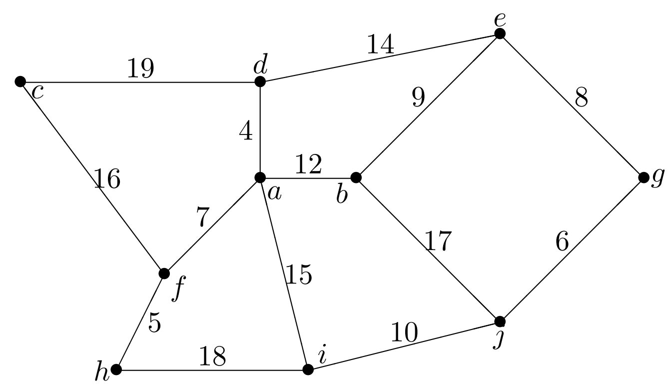
Round	Vertex	s	a	b	c	d	e
0							
1							
2							
3							
4							
5							
6							

b) Draw the "previous pointers" or "back edges" found by your simulation of Dijkstra's algorithm.

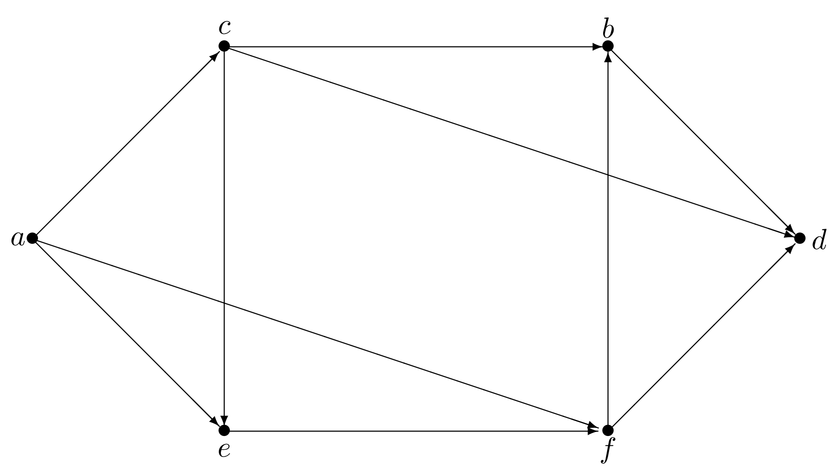


Problem 2. Algorithm Simulation (10 points):

a) Consider the following undirected graph. List the order of edges added to the minimum spanning tree by an execution of Kruskal's algorithm.



b) Consider the following directed graph. List the vertices in an order that they could be picked by a topological sort algorithm. (In other words, list the vertices in topological order.)



Problem 3. Changing Edge Costs (10 points):

a) *True or false*: Adding one to the cost of every edge in the graph in the shortest paths problem never changes the resulting shortest paths. To be more precise: Let $G = (V, E)$ be a directed graph with edge costs $c(e)$. Let $G' = (V, E)$ be a directed graph with edge costs with the same vertices and edges, with $c'(e) = c(e) + 1$. *True or false*: If P is a shortest path between vertices s and t in G, then P is necessarily a shortest path between s and t in G'. Justify your answer.

b) *True or false*: Adding one to the cost of every edge in the graph in the minimum spanning tree problem does not change the resulting minimum spanning tree. To be more precise: Let $G = (V, E)$ be a connected, undirected graph with edge costs $c(e)$. Let $G' = (V, E)$ be an undirected graph with edge costs with the same vertices and edges, with $c'(e) = c(e)+1$. *True or false*: If T is a minimum spanning tree for G, then T is necessarily a minimum spanning tree for G'. Justify your answer.

Problem 4. Stable Matching (10 points):

Consider the following instance of the stable matching problem. The preference lists are:

$$M = \begin{bmatrix} m_1: & w_1 & w_2 & w_3 & w_4 \\ m_2: & w_1 & w_2 & w_4 & w_3 \\ m_3: & w_2 & w_4 & w_3 & w_1 \\ m_4: & w_2 & w_1 & w_3 & w_4 \end{bmatrix}$$

$$W = \begin{bmatrix} w_1: & m_1 & m_4 & m_3 & m_2 \\ w_2: & m_2 & m_1 & m_4 & m_3 \\ w_3: & m_3 & m_4 & m_1 & m_2 \\ w_4: & m_1 & m_2 & m_4 & m_3 \end{bmatrix}$$

True or false: the pair (m_2, w_2) is matched in every stable matching for this instance. Justify your answer. (Note: this instance of the stable matching problem has more than one possible solution.)

Problem 5. Interval Scheduling (10 points):

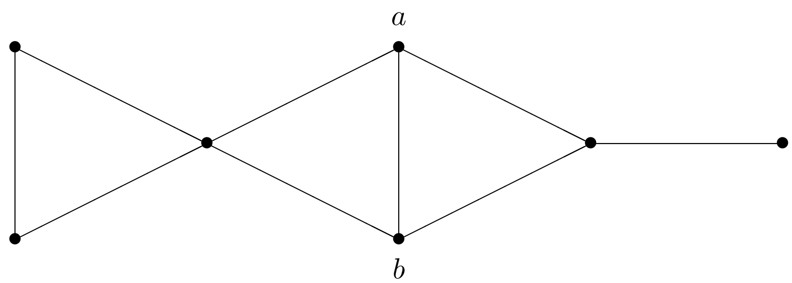
The input for an interval scheduling problem is a set of intervals $I = \{i_1, \ldots, i_n\}$ where i_k has start time s_k, and finish time f_k. The problem is to find a set of *non-overlapping intervals* that satisfies a given criteria. The greedy algorithm, with respect to an ordering $\mathcal{O}$ iterates through the intervals in the order given by $\mathcal{O}$, and adds intervals to the solution that are compatible (e.g., non-intersecting) with intervals already placed in the solution.

a) Suppose that you want to maximize the cumulative length of the selected intervals. *True or false*: The greedy algorithm based on selecting intervals in order of increasing start time finds an optimal solution. Justify your answer.

b) Suppose that all intervals have the same length, and you want to maximize the number of selected intervals. *True or false*: The greedy algorithm based on selecting intervals in order of increasing start time finds an optimal solution. Justify your answer.

Problem 6. Centroids (10 points):

Let $G = (V, E)$ be a connected, undirected graph. Let $d(a, b)$ be the shortest path distance from a to b. We define $MaxD(a)$ to be the maximum of the shortest path distances from a to the other vertices in the graph, i.e., $MaxD(a) = \max_v d(a, v)$. A vertex is said to be a *centroid* if it attains the minimum of $MaxD$ over all vertices. Note that there can be multiple centroids. In the graph below, the vertices labeled a and b are centroids.



Describe an algorithm for finding a centroid in a connected graph. Given a connected, undirected graph $G = (V, E)$, your algorithm should return a vertex v which is one of the centroids of the graph. You can give a high level description, and you can use algorithms described in class (without reproducing the details). The runtime of your algorithm should be $O(nm)$. You should describe why your algorithm computes a centroid and also give and justify the algorithm's run time.