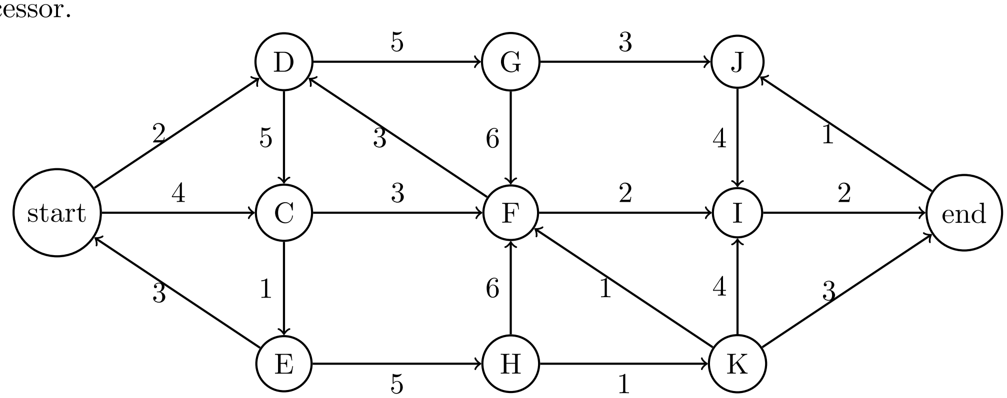
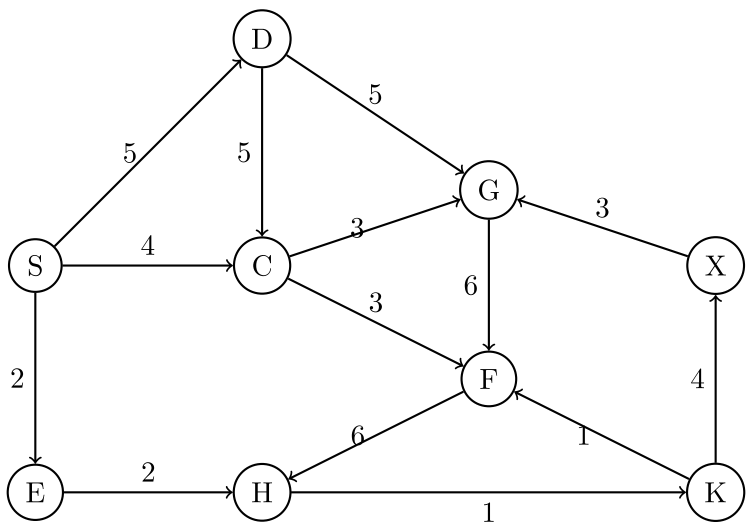
CSC-361: Extra Graphs Problems: Path-Finding and Minimum Spanning Trees

1. Consider a path from node (start) to node (end). Apply Dijkstra's algorithm to identify (1) the length of the shortest path and (2) the shortest path. State the shortest path you identified and its length explicitly. For your final answer: (1) annotate ALL nodes with their shortest path values and (2) indicate the predecessor node with an arrow from a node to its predecessor.



Problems 2 and 3 refer to the following graph:



2. We will define an ordering rule here that *whenever given a choice, choose the smallest node alphabetically.* Identify all backedges in the graph.

3. Omitting the backedges you identified in question 2 from the original graph, identify a shortest path in the resulting DAG from node (S) to node (X). You are not allowed to use Dijkstra's algorithm.

   For your solution, (a) redraw the graph so the nodes are aligned linearly according to a topological sort (as depicted in the courses videos and slides). Then (b) execute the search algorithm displaying the *final* distances from the start node to each node in the DAG. Show all work.

4. Indicate true / false for each statement.

   (a) Let G be an undirected, weighted graph with a corresponding minimum spanning tree T. Consider scaling the weights in G by adding to each weight a constant $k > 0$. T is still a minimum spanning tree of G with rescaled weights.

(b) Let G be an undirected, weighted graph with a corresponding minimum spanning tree T. Consider scaling the weights in G by multiplying each weight by a constant $k > 0$. T is still a minimum spanning tree of G with rescaled weights.

(c) An undirected, weighted graph has a unique minimum spanning tree if the edge weights are unique.

(d) You can compute a maximum spanning tree by negating the weights in an undirected, weighted graph and executing Kruskal's algorithm.

5. Find a minimum spanning tree for each of the graphs shown below.