NOVA SCHOOL OF BUSINESS & ECONOMICS

2494 - COMPUTATIONAL THINKING & DATA SCIENCE

2021-22, Spring Semester

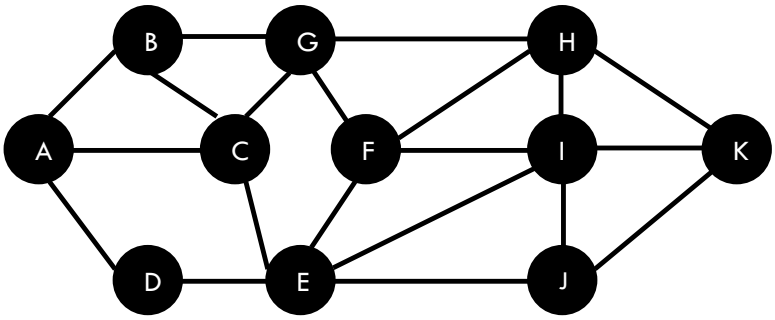
In-class Exercises

GRAPH-THEORETIC MODELS

1. Consider an unweighted graph G = (V, E), where V is the set of nodes and E the set of edges. Suppose that you want to determine the shortest path from node ***s*** to node ***t***, but if it is possible you would like to stop by node ***k*** in the way. However, you only want to pass by node ***k*** if the length of your path doesn't increase more than δ%.

   Write a Python program that helps you to determine the shortest path between ***s*** and ***t*** given your preference for stopping at node ***k***.

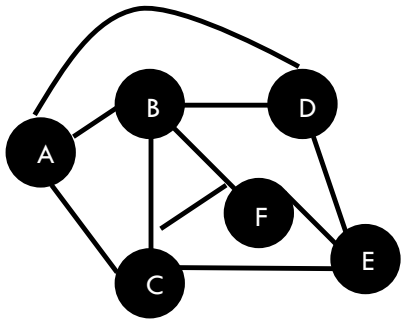
   Note: It should either return the shortest path from ***s*** to ***t*** or the shortest path from ***s*** to ***t*** containing ***k***, depending on the situation.



   Use the following graph to help you testing your program.

   Consider ***s*** = A, ***t*** = K and ***k*** = F.

2. Write the sequence of visited nodes that corresponds to the one obtained through the application of the depth first search to the graph given below. The search starts at node A and lexicographic ordering is assumed for the edges emanating from each node.



3. Consider a graph G. Let d(a,b) denote the length of the shortest path between the nodes a and b.
   Assuming that v is visited before u in the breadth first search algorithm starting on node r, compare (>, = or <), if possible, the value of d(r,v) with the value of d(r,u).