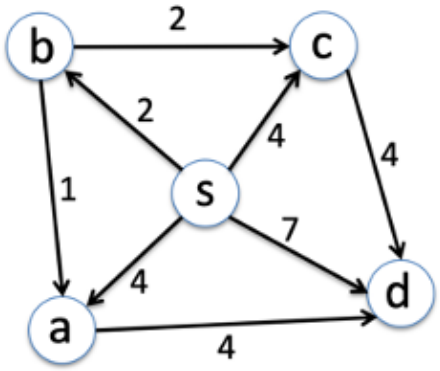
CMSC 401 SPRING 2022 – Theory Homework Assignment 2 (ThHW2)

PROBLEM 1

Analyze iterations of Dijkstra shortest path method for the graph below, with vertex **s** as the source.



What are the temporary distances to each node AFTER each iteration? For each iteration, for each node, provide the value of the temporary distance. Assume that before the first iteration, the temporary distances are: 0 for node s, +infinity for all other nodes.

SOLUTION:

node iteration	s	a	b	c	d
Before 1st	0	+INF	+INF	+INF	+INF
After 1st					
After 2nd					
After 3rd					
After 4th					
After 5th					

PROBLEM 2

Imagine the following problem:

The corporation you work for has just established a number of offices in a foreign land. But the CEO is worried that corporate strategies and trade secrets will be exposed if regular telecommunication channels like the internet are used for communication between these offices. After a long brain-storming session, the Board of Directors came up with the solution: pigeons.

By nature, pigeons know how to fly to wherever their nest is established. They can also be trained to fly back to a second location, where their food is regularly delivered. Thus, messenger pigeons are an effective, hard-to-intercept, point-to-point communication link. To link two offices, establish a pigeon nest in one office, feed that pigeon in the other office, and it will learn to fly between the two offices and carry some messages. The only problem is that pigeons become unreliable over long distances, and the corporation needs 100% reliability.

You, as the Director of IT, have been charged with verifying if the plan will work. You consulted biology experts and learned that for distances of up to 200 miles, 100% reliability can be achieved by well-trained pigeons. Above 200 miles the reliability drops below 100%, which is not acceptable. Given the geographic locations of the offices, you calculated direct straight-line distances between all pairs of offices.

Based on all the information you gathered, you need to decide whether any office can deliver a message to any other office, with 100% reliability, via pigeon post (possibly using a pigeon relay involving multiple office-to-office hops, with each hop at most 200 miles). You asked your IT team to submit possible options on how to solve the problem, and you received the following options:

A) *Create a graph with one node per office. For each pair of offices, look up the direct straight-line distance between them, in miles. Add an edge between each pair of offices, with the distances as edge weights. Run MST on the graph. Inspect the MST, if all edges in it are below 200 miles, return "yes, pigeon post will work". If the tree contains one or more edges with distance above 200 miles, return "no, cannot achieve 100% reliability".*
B) *Create a graph with one node per office. For each pair of offices, look up the direct straight-line distance between them, in miles. If the distance is 200 miles or less, add an edge between the pair of offices to the graph, if the distance is above 200 miles, no edge. Run all-pairs shortest path algorithm. If at the end all the vertex-vertex shortest distances in the graph are finite return "yes, pigeon post will work". If at least one pair has "+INF", return "no, cannot achieve 100% reliability".*
C) *Create a graph with one node per office. For each pair of offices, look up the direct straight-line distance between them, in miles. Add an edge between each pair of offices, with the distances as edge weights. Run all-pairs shortest path algorithm on the graph. Inspect the shortest paths between every pair of nodes, if all the edges on*

all the paths are 200 miles or less, return "yes, pigeon post will work", if there is one or more edge with weight greater than 200 miles, return "no, cannot achieve 100% reliability".

D) *Create a graph with one node per office. For each pair of offices, look up the direct straight-line distance between them, in miles. If the distance is 200 miles or less, add an edge between the pair of offices to the graph, if the distance is above 200 miles, no edge. Detect if the graph has only one connected component, or more than one: run depth-first search complexity starting from a random node. If the graph has a single connected component (all nodes were visited by the DFS without any restarts), then answer "yes, pigeon post will work", otherwise answer "no, cannot achieve 100% reliability".*

E) *Create a graph with one node per office. For each pair of offices, look up the direct straight-line distance between them, in miles. If the distance is 200 miles or less, add an edge between the pair of offices to the graph, if the distance is above 200 miles, no edge. Run MST on the graph. If the MST exists (all nodes can be connected by a tree with edges <=200), answer "yes, pigeon post will work". If the MST algorithm fails (cannot reach/connect some nodes), answer "no, cannot achieve 100% reliability".*

F) *Create a graph with one node per office. For each pair of offices, look up the direct straight-line distance between them, in miles. If the distance is 200 miles or less, add an edge between the pair of offices to the graph, if the distance is above 200 miles, no edge. Run single-source shortest path algorithm from a random node. If at the end some vertices have "+INF" distance, they can't be reached, answer "no, cannot achieve 100% reliability". If all vertices have finite distance, answer "yes, pigeon post will work".*

Analyze the 6 options above, and for each, answer whether it will lead to correct results, and what is its computational complexity. Which option is the best one?

Option	Correct?	Computational complexity?
A		
B		
C		
D		
E		
F		

Best option: