

Question 1: We're still working through this transportation logistics scenario. Once again, suppose every route has an associated cost, and now, let's suppose that the cost of going from city A to city B is the same as going from city B to city A. Management has inflicted another cost cutting measure on us -- we need to cancel as many routes as we can, but still make sure that our goods can get from any one city to another. We want to do this, but we want to make sure that the total cost of all routes is as small as possible. What algorithm do we use?

- **A:** Depth-First Search
- **B:** Breadth-First Search
- **C:** Dijkstra's Algorithm
- **D:** Network Flow
- **E:** Dynamic Programming
- **F:** Minimum Spanning Tree
- **G:** Topological Sort

Question 2: If I have C cities and R routes, what is the running time?

Question 3: The management of Question 1 has been fired, and we go back to having all of the original routes available to us. A terrorist group has decided that they want to make it impossible to have goods go from Bangor to San Diego. The first thing they will do is bomb all of the airports, so no more plane travel. Then, for each train route, they have determined the cost of bombing that route. What algorithm should they use to figure out which train routes to destroy that will achieve their goals in the cheapest manner? (Use the answers from Question 1).

Question 4: The terrorist group has done their bombings. Some of them have been successful, and some of them haven't. You work for the government and know all of the success or failure of each bombing. You are asked if it's possible to get goods from Seattle to Miami. Which algorithm do you use to make this determination?

Question 1: Minimum spanning tree: **F**.

Question 2: $O(R \log C)$.

Question 3: They are trying to find the minimum cut, which you find with network flow: **D**.

Question 4: You're finding the path between two nodes in a graph. That's DFS: **A**.
