

Question 1: I am the head of a transportation logistics company. We have two divisions -- air and train. Each of these divisions keeps the following information on cities and the routes between them. Cities are endpoints, where we can store goods. Routes may exist between two cities in either of the divisions. For each route, there is a time to travel the route, and a cost that it takes to travel the route.

Suppose I need to get goods from city A to city B, and I want to do it in cheapest manner, money-wise. I know there will be a way to go from A to B. Which algorithm should I use? Choose from the following:

- **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: A new management regime takes over, and dictates that every plane or train trip will cost \$1,000. I need to get goods from city A to city B, and I want to do it in cheapest manner. Which algorithm should I use (Use the answers from Question 1)?

Question 4: If I have C cities and R routes, what is the running time?

Question 5: The old management regime takes over again, and the cost for each route has returned to its original cost from Question 1. Suppose that all of the routes travel south and west, and I want to get my goods from Bangor, Maine to San Diego, California. I want to do this in the cheapest manner possible. In terms of Big-O, what algorithm should I use? Use the answers from Question 1.

Question 6: If I have C cities and R routes, what is the running time?

Question 1: Although the problem is framed in terms of "cost", this is a shortest path problem. The answer is **C**, Dijkstra's Algorithm.

Question 2: Dijkstra's algorithm will be $O(R \log C)$.

Question 3: Since the cost of each route is the same, we need to minimize the number of routes we take. That's shortest path on an unweighted graph: Breadth-First Search: **B**.

Question 4: Breadth-first search is $O(C+R)$, and I think this is the proper answer. Although I say "I know there will be a way to go from A to B," does that mean for every A and every B? I don't think so, so $O(C+R)$ it is. If I said "all of the cities are connected," then the proper answer is $O(R)$.

Question 5: When all routes go south and west, the graph becomes acyclic, which means that Topological Sort will apply: **G**.

Question 6: $O(C+R)$.