Algorithm Explanation

BFS(Breadth First Algorithm) is an algorithm for traversing an unweighted graph for a tree. It start with a root node and explores each adjacent node before exploring nodes at the next level. It uses the queue for storing the visited nodes of the graph/tree.

As per the assignment we have to find the route between two cities that is between source and the destination.

The function "findAnyRouteToCity" will run successfully only if the route from source to destination is smoothly reachable.

In this algorithm of traversing I am using BFS to solve the problem statement which are described below:

- ➢ First, we will create a HashMap named "visitedNode" which stores the value true or false along with the nodes indicating that the node is visited or not.
- ➢ Next, we will create an another HashMap named "routeMap" which will store the path from source to destination.
- ➢ Now, we will create an empty queue for traversing purpose.
- ➢ In next step, initializing the visited Node HashMap to false for all cities.
- ➢ Now, initializing the source to true in visited Node HashMap.
- ➢ Add the source to queue.
- ➢ Till the queue is empty:

  - ✓ After removing top element from queue we are storing it in current city.
  - ✓ Now, we are checking all the cities connected to current city and checking that if it is visited or not.
  - ✓ If the city is not visited than add into queue.
  - ✓ After adding update the false value of city to true in visited Node map.
  - ✓ Now, add the route to the route HashMap.

- If the destination is not visited or the value of destination is false in visited Node map than display the city is “Not Reachable”.

- Else, initialize the current city to destination and traverse through the route HashMap while current city is not equal to source add cities to path and at last print the route.

Thanku