Problem Statement

PART A

You are in charge of the security of Prime Minister Modi, who is going on a tour across India. Prime Minister tells y ou the name of the next city he wants to go, and you need to inform him whether it is possible to reach that city or n ot from a particular city, if it is possible, return the route between the cities.

To find the reachability from one specific city to another, you need to design a java program to read the pair of conn ected city names from the standard input in the following format,

Mumbai,Pune

Pune,Indore

Jaipur,Delhi

The given input information of connected cities is stored as a graph.

Now, write a function findAnyRouteToCity(String source, String destination) which uses the graph above to return a path if possible between the source and destination, or "Not Reachable" in case no such path exists.

For example: Consider the above input information as a graph,

findAnyRouteToCity("Mumbai", "Indore")

Mumbai -> Pune -> Indore

findAnyRouteToCity("Mumbai", "Jaipur")

Not Reachable

Also, justify the traversal algorithm chosen between DFS and BFS to solve the problem statement.

Note: You are only allowed to use the following java packages,

java.util.HashMap
java.util.HashSet
java.util.LinkedList
java.util.Set
java.util.Map
java.util.Queue
java.util.Scanner
Sample Input:

Enter the Map
Mumbai,Pune
Pune,Indore
Jaipur,Delhi

Enter the source
Mumbai
Enter the destination
Indore

Sample Output:
The route from Mumbai to Indore is
Mumbai -> Pune -> Indore

PART B

Your next task is to ensure that the path taken by the prime minister is as safe as possible. You send out your team members to explore all the roads and they report values from 1-100 depending on how unsafe the road is. With this data, you need to return the safest route that should be taken by the prime minister, where the overall safety of the path is defined by the sum of the safeties of the roads.

Also, if you don't get a report for a particular path from your team, then that path needs to be avoided at all costs because it is highly unsafe.

In order to find the safest route, you need to design a java program to read the pair of connected city names with the report values as a standard input in the following format,

Mumbai,Pune,10

Pune,Indore,60

Jaipur,Delhi,5

Mumbai,Thane,99

Thane,Indore,80

The given input information of connected cities with their report values is stored as a graph.

Now, write a function findSafestRouteToCity(String source, String destination) which uses the graph to return the sa fest route between source and destination, or "Not Recommended" in case no path exists.

For example: Consider the above input information as a graph,

findSafestRouteToCity("Pune", "Jaipur")

Not Recommended

findSafestRouteToCity("Indore", "Mumbai")

Indore -> Pune -> Mumbai

Note: (Indore, Pune, Mumbai) {Safety score: 60+10} is safer than (Indore, Thane, Mumbai) {Safety score: 80+99}.

Also, keep in mind that terrorists have hacked into Java packages, and you can only import these packages which are marked as safe:

java.util.Comparator
java.util.HashMap
java.util.HashSet
java.util.Set
java.util.Map
java.util.PriorityQueue
java.util.Scanner
Sample Input:

Enter the graph
Mumbai,Pune,10
Pune,Indore,60
Jaipur,Delhi,5
Mumbai,Thane,99
Thane,Indore,80

Enter the source
Indore
Enter the destination
Mumbai

Sample Output:

The route from Indore to Mumbai is
Indore -> Pune -> Mumbai