PIMPRI CHINCHWAD EDUCATION TRUST'S

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

Department of Computer Engineering

Mini Project Report

On

"Airline Routing System"

Subject: Design and Analysis of Algorithm

Academic Year: 2023-24

Semester: I

Submitted by	**Name of the student**	**Roll number**
	1. Sakshi Sharad Kulkarni	TYCOC289
	2. Janhavi Vijay Mali	TYCOC292
	3. Vinayak Madan Shete	TYCOC303

Submitted to: Prof. S . D. Rajput

Date: 07/11/2023

Sign:

Project Title: Airline Routing System

Problem statement:

Design and implement an Airline Routing System that utilizes the All-Pair Shortest Path algorithm to optimize flight paths, ensuring efficient connectivity between various airports served by an airline. The system should consider multiple factors, such as flight distances, fuel efficiency, time constraints, and potential constraints such as weather conditions or airspace limitations.

Introduction:

The Floyd-Warshall algorithm is a dynamic programming technique used to find the shortest paths in a weighted graph with positive or negative edge weights. Unlike Dijkstra's algorithm, which focuses on finding the shortest path between a single source and all other vertices, the Floyd-Warshall algorithm calculates the shortest path between all pairs of vertices in a graph.

Approach to Solution:

Algorithm:

1. **Initialization:** Create a 2D array (let's call it distance) to store the shortest distances between all pairs of vertices. Initially, this array is filled with the direct edge weights between vertices. If there's no direct edge between two vertices, the distance is considered infinity. Also, set the diagonal elements of the matrix to zero (distance from a vertex to itself).

2. **Main Calculation:** Perform a series of comparisons and updates to gradually improve the shortest path estimates. Consider each pair of vertices (i, j) and each possible intermediate vertex k.

   For each pair (i, j), if the distance from i to j through vertex k is shorter than the current distance, update the distance array with this shorter distance.

3. **Optimization:** Repeat the process for all vertices as intermediate points, iteratively improving the shortest path estimates.

4. **Final Output:** The distance array will contain the shortest path distances between all pairs of vertices when the algorithm finishes.

Complexity:

Time complexity: O(V^3), where V is the number of vertices.

Space complexity: O(V^2), since it requires a 2D array to store distances.

The Floyd-Warshall algorithm is versatile and can handle graphs with negative weights, but it's not suitable for large graphs due to its time complexity. However, for small graphs or when the goal is to find all pairs shortest paths, it can be quite effective.

Input Code:

```
import java.util.*;

class Airport
{
	private final String code;
	private final String name;

	public Airport(String code, String name)
	{
		this.code = code;
		this.name = name;
	}

	public String getCode()
	{
		return code;
	}

	public String getName()
	{
		return name;
	}
}

class FlightRoute
{
	private final Airport source;
	private final Airport destination;
	private final int distance;

	public FlightRoute(Airport source, Airport destination, int distance)
	{
		this.source = source;
		this.destination = destination;
		this.distance = distance;
	}
```

```
	public Airport getSource()
	{
		return source;
	}




	public Airport getDestination()
	{
		return destination;
	}

	public int getDistance()
	{
		return distance;
	}
}

public class FlightRoutingSystem
{
	private final Map<String, Airport> airports;
	private final List<FlightRoute> routes;

	public FlightRoutingSystem()
	{
		airports = new HashMap<>();
		routes = new ArrayList<>();
	}

	public void addAirport(String code, String name)
	{
		airports.put(code, new Airport(code, name));
	}

	public  void  addRoute(String  sourceCode,  String  destinationCode,  int
distance)
	{
		Airport source = airports.get(sourceCode);
		Airport destination = airports.get(destinationCode);
		if (source != null && destination != null)
		{
			routes.add(new FlightRoute(source, destination, distance));
	}
 }

public    List<List<Airport>>    findAllRoutes(String    sourceCode,    String
destinationCode)
{
	Airport source = airports.get(sourceCode);
	Airport destination = airports.get(destinationCode);

	if (source == null || destination == null)
	{
		return null; // Source or destination airport not found
	}
```

```
      List<List<Airport>> allRoutes = new ArrayList<>();
      List<Airport> currentRoute = new ArrayList<>();
      findRoutesDFS(source, destination, allRoutes, currentRoute);

      return allRoutes;
}




private   void   findRoutesDFS(Airport   currentAirport,   Airport   destination,
List<List<Airport>> allRoutes,
   List<Airport> currentRoute)
{
      currentRoute.add(currentAirport);

      if (currentAirport == destination)
      {
            allRoutes.add(new ArrayList<>(currentRoute));
      }
      else
      {
            for (FlightRoute route : routes)
            {
                  if      (route.getSource()      ==      currentAirport      &&
!currentRoute.contains(route.getDestination()))
                  {
                  findRoutesDFS(route.getDestination(),  destination,  allRoutes,
currentRoute);
                  }
            }
      }
      currentRoute.remove(currentRoute.size() - 1);
}

public void displayAllRoutes(List<List<Airport>> allRoutes)
{
      if (allRoutes != null && !allRoutes.isEmpty())
      {

      System.out.println("\n==================================================");
            System.out.println("All Available Routes are:");

            List<Airport> shortestRoute = null;
            int shortestDistance = Integer.MAX_VALUE;

            for (List<Airport> route : allRoutes)
            {
                  int totalDistance = 0;
                  System.out.print(route.get(0).getCode());

                  for (int i = 1; i < route.size(); i++)
                  {
                        for (FlightRoute flightRoute : routes)
```

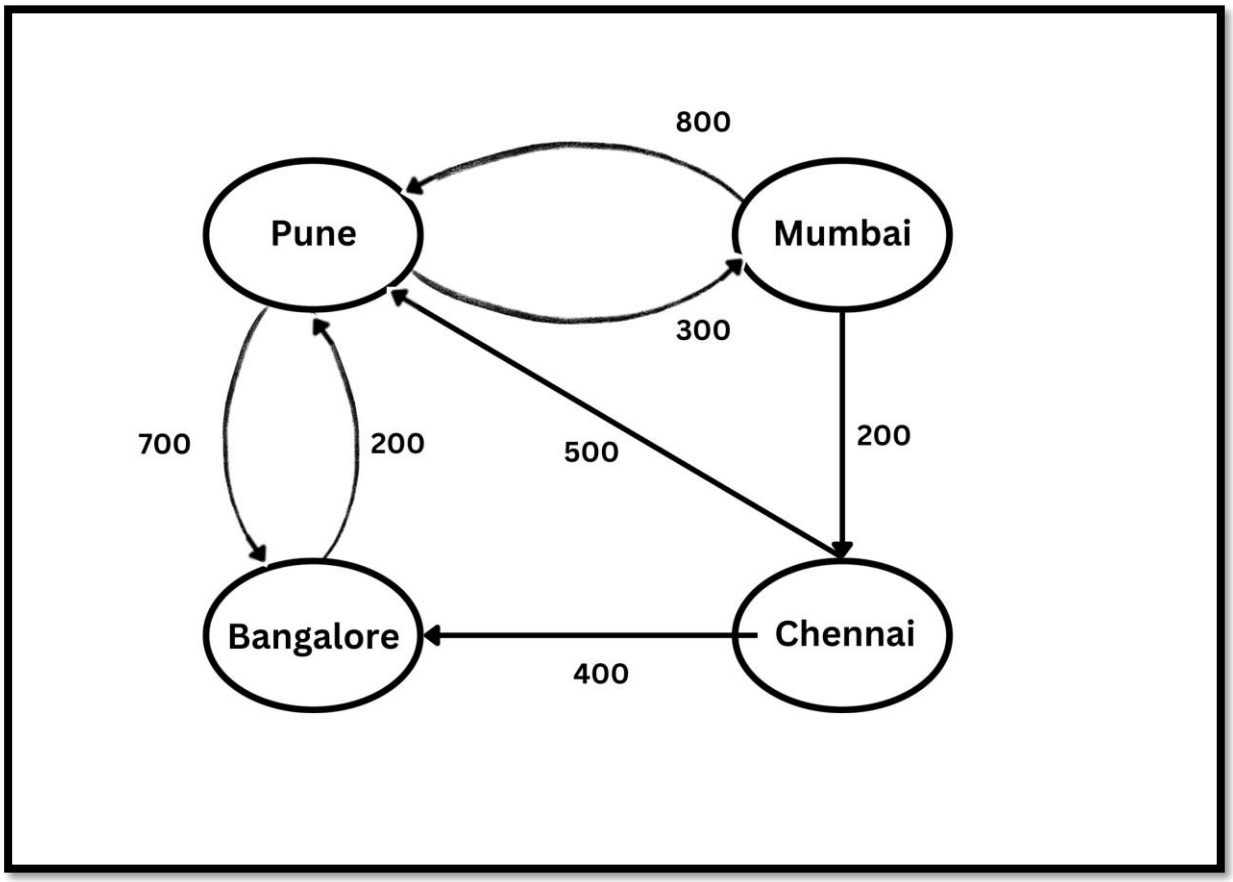
```
                        {
                                if (flightRoute.getSource() == route.get(i -
1) && flightRoute.getDestination() ==
      route.get(i))
                                {
                                        totalDistance                        +=
flightRoute.getDistance();
                                        System.out.print("     to     "     +
route.get(i).getCode());
                                        break;
                                }
                        }




                }

                System.out.println(" Distance: " + totalDistance);

                if (totalDistance < shortestDistance)
                {
                      shortestDistance = totalDistance;
                      shortestRoute = route;
                }

      System.out.println("\n================================================");
      }

      System.out.println("\nShortest Route:");
      if (shortestRoute != null)
      {

      System.out.println("\n================================================");
            for (Airport airport : shortestRoute)
            {
                  System.out.println(airport.getCode()    +    "    -    "    +
airport.getName());
            }
            System.out.println("Distance: " + shortestDistance);

      System.out.println("\n================================================");
      }
      else
      {
            System.out.println("No routes found.");
      }
       }
      else
      {
            System.out.println("No routes found.");
      }
}

public void displayAirports()
{
      System.out.println("\n================================================");
```

```
        System.out.println("Airports:");
        for (Airport airport : airports.values())
        {
                System.out.println(airport.getCode() + " - " + airport.getName());
        }
        System.out.println("\n================================================");
}

public void displayRoutes()
{
        System.out.println("\n================================================");
        System.out.println("Flight Routes:");
        for (FlightRoute route : routes)
        {




System.out.println(route.getSource().getCode()      +      "      to      "      +
route.getDestination().getCode() + " Distance: " +
        route.getDistance());
        }
        System.out.println("\n================================================");
}

public void displayShortestRoute(List<Airport> shortestRoute)
{
        if (shortestRoute != null)
        {

        System.out.println("\n================================================");
                System.out.println("Shortest Route:");
                for (Airport airport : shortestRoute)
                {
                        System.out.println(airport.getCode()    +    "    -    "    +
airport.getName());
                }

        System.out.println("\n================================================");
        }
        else
        {
                System.out.println("No route found.");
        }
}

public static void main(String[] args)
{
        FlightRoutingSystem routingSystem = new FlightRoutingSystem();
        Scanner scanner = new Scanner(System.in);

        while (true)
        {

        System.out.println("\n================================================");
                System.out.println("\nFlight Routing System Menu:");
```

```
            System.out.println("1. Add Airport");
            System.out.println("2. Add Flight Route");
            System.out.println("3. Find Shortest Route");
            System.out.println("4. Display Airports");
            System.out.println("5. Display Flight Routes");
            System.out.println("6. Exit");

      System.out.println("\n================================================");
            System.out.print("Select an option: ");

            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            switch (choice)
            {
                  case 1:

      System.out.println("\n================================================");
                        System.out.println("\n==================ADD
AIRPORT=================");



      System.out.println("\n================================================");
                        System.out.print("Enter Airport Code: ");
                        String airportCode = scanner.nextLine();
                        System.out.print("Enter Airport Name: ");
                        String airportName = scanner.nextLine();
                        routingSystem.addAirport(airportCode, airportName);
                        System.out.println("Airport added: " + airportCode);
                        break;

                  case 2:

      System.out.println("\n================================================");
                        System.out.println("\n================ADD         FLIGHT
ROUTE===============");

      System.out.println("\n================================================");
                        System.out.print("Enter Source Airport Code: ");
                        String sourceCode = scanner.nextLine();
                        System.out.print("Enter Destination Airport Code: ");
                        String destinationCode = scanner.nextLine();
                        System.out.print("Enter Distance: ");
                        int distance = scanner.nextInt();
                        routingSystem.addRoute(sourceCode,       destinationCode,
distance);
                        System.out.println("Flight Route added: " + sourceCode +
" to " + destinationCode);
                        break;

                  case 3:

      System.out.println("\n================================================");
                        System.out.println("\n================FIND      SHORTEST
ROUTE============");

      System.out.println("\n================================================");
```

```
                        System.out.print("Enter Source Airport Code: ");
                        String sourceAirportCode = scanner.nextLine();
                        System.out.print("Enter Destination Airport Code: ");
                        String destinationAirportCode = scanner.nextLine();
                        List<List<Airport>>              allRoutes              =
routingSystem.findAllRoutes(sourceAirportCode, destinationAirportCode);
                        routingSystem.displayAllRoutes(allRoutes);
                        break;

                  case 4:

      System.out.println("\n================================================");
                        System.out.println("\n================DISPLAY
AIRPORTS================");

      System.out.println("\n================================================");
                        routingSystem.displayAirports();
                        break;

                  case 5:

      System.out.println("\n================================================");
                        System.out.println("\n================DISPLAY
ROUTES==================");




      System.out.println("\n================================================");
                        routingSystem.displayRoutes();
                        break;

                  case 6:
                        System.out.println("Exiting Flight Routing System.");
                        System.exit(0);

                  default:
                        System.out.println("Invalid option. Please select a valid
option.");
            }
      }
}
}


// Input==>
/*
 * Pune->Mumbai=120
 * Mumbai->Pune=140
 * Pune->Banglore=800
 * Pune->Chennai=500
 * Mumbai->chennai=700
 * chennai->banglore=200
 * mumbai->banglore=600
 *
 *
 */
```

Ouput:




```
C:\Windows\System32\cmd.exe - java FlightRoutingSystem

D:\PCCOE\Semester5\DAA>javac FlightRoutingSystem.java

D:\PCCOE\Semester5\DAA>java FlightRoutingSystem

================================================

Flight Routing System Menu:
1. Add Airport
2. Add Flight Route
3. Find Shortest Route
4. Display Airports
5. Display Flight Routes
6. Exit

================================================
Select an option:
```

Adding Airports:

```
D:\PCCOE\Semester5\DAA>javac FlightRoutingSystem.java

D:\PCCOE\Semester5\DAA>java FlightRoutingSystem

==================================================

Flight Routing System Menu:
1. Add Airport
2. Add Flight Route
3. Find Shortest Route
4. Display Airports
5. Display Flight Routes
6. Exit

==================================================
Select an option: 1

==================================================

====================ADD AIRPORT==================

==================================================
Enter Airport Code: PUN
Enter Airport Name: Pune
Airport added: PUN

==================================================

Flight Routing System Menu:
1. Add Airport
2. Add Flight Route
3. Find Shortest Route
4. Display Airports
5. Display Flight Routes
6. Exit

==================================================
Select an option:
```

```
====================ADD AIRPORT==================

==================================================
Enter Airport Code: MUM
Enter Airport Name: Mumbai
Airport added: MUM

==================================================

Flight Routing System Menu:
1. Add Airport
2. Add Flight Route
3. Find Shortest Route
4. Display Airports
5. Display Flight Routes
6. Exit
```

```
==================================================

====================ADD AIRPORT=================

==================================================
Enter Airport Code: CHN
Enter Airport Name: Chennai
Airport added: CHN

==================================================

Flight Routing System Menu:
1. Add Airport
2. Add Flight Route
3. Find Shortest Route
4. Display Airports
5. Display Flight Routes
6. Exit

==================================================
Select an option: 1

==================================================

====================ADD AIRPORT=================

==================================================
Enter Airport Code: BNG
Enter Airport Name: Bangalore
Airport added: BNG

==================================================
```

Displaying Airports:

```
Flight Routing System Menu:
1. Add Airport
2. Add Flight Route
3. Find Shortest Route
4. Display Airports
5. Display Flight Routes
6. Exit

==================================================
Select an option: 4

==================================================

================DISPLAY AIRPORTS================

==================================================

==================================================
Airports:
MUM - Mumbai
PUN - Pune
CHN - Chennai
BNG - Bangalore

==================================================

==================================================
```

Adding Flight Routes:

```
================================================

Flight Routing System Menu:
1. Add Airport
2. Add Flight Route
3. Find Shortest Route
4. Display Airports
5. Display Flight Routes
6. Exit

================================================
Select an option: 2

================================================

=================ADD FLIGHT ROUTE================

================================================
Enter Source Airport Code: MUM
Enter Destination Airport Code: PUN
Enter Distance: 800
Flight Route added: MUM to PUN

================================================

Flight Routing System Menu:
1. Add Airport
2. Add Flight Route
3. Find Shortest Route
4. Display Airports
```

```
Flight Routing System Menu:
1. Add Airport
2. Add Flight Route
3. Find Shortest Route
4. Display Airports
5. Display Flight Routes
6. Exit

================================================
Select an option: 2

================================================

=================ADD FLIGHT ROUTE================

================================================
Enter Source Airport Code: MUM
Enter Destination Airport Code: CHN
Enter Distance: 200
Flight Route added: MUM to CHN

================================================
```

```
================================================
Select an option: 2

================================================

=================ADD FLIGHT ROUTE================

================================================
Enter Source Airport Code: CHN
Enter Destination Airport Code: PUN
Enter Distance: 500
Flight Route added: CHN to PUN

================================================

Flight Routing System Menu:
1. Add Airport
2. Add Flight Route
3. Find Shortest Route
4. Display Airports
5. Display Flight Routes
6. Exit
```

Displaying Routes:

```
================================================

Flight Routing System Menu:
1. Add Airport
2. Add Flight Route
3. Find Shortest Route
4. Display Airports
5. Display Flight Routes
6. Exit

================================================
Select an option: 5

================================================

=================DISPLAY ROUTES=================

================================================

================================================
Flight Routes:
PUN to MUM Distance: 300
MUM to PUN Distance: 800
PUN to BNG Distance: 700
BNG to PUN Distance: 200
MUM to CHN Distance: 200
CHN to BNG Distance: 100
CHN to PUN Distance: 500

================================================

================================================
```

Finding Shortest Path:

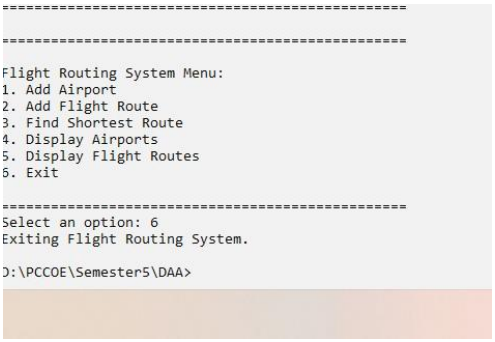
```
================================================

=================FIND SHORTEST ROUTE=============

================================================
Enter Source Airport Code: MUM
Enter Destination Airport Code: BNG

================================================
All Available Routes are:
MUM to PUN to BNG Distance: 1500

================================================
MUM to CHN to BNG Distance: 300

================================================
MUM to CHN to PUN to BNG Distance: 1400

================================================

Shortest Route:

================================================
MUM - Mumbai
CHN - Chennai
BNG - Bangalore
Distance: 300

================================================

================================================
```

```
================================================
Select an option: 3

================================================

=================FIND SHORTEST ROUTE=============

================================================
Enter Source Airport Code: BNG
Enter Destination Airport Code: MUM

================================================
All Available Routes are:
BNG to PUN to MUM Distance: 500

================================================

Shortest Route:

================================================
BNG - Bangalore
PUN - Pune
MUM - Mumbai
Distance: 500

================================================

================================================
```

```
================================================
Select an option: 3

================================================

=================FIND SHORTEST ROUTE=============

================================================
Enter Source Airport Code: BNG
Enter Destination Airport Code: CHN

================================================
All Available Routes are:
BNG to PUN to MUM to CHN Distance: 700

================================================

Shortest Route:

================================================
BNG - Bangalore
PUN - Pune
MUM - Mumbai
CHN - Chennai
Distance: 700

================================================

================================================
```

```
================================================
Select an option: 3

================================================

=================FIND SHORTEST ROUTE=============

================================================
Enter Source Airport Code: BNG
Enter Destination Airport Code: CHN

================================================
All Available Routes are:
BNG to PUN to MUM to CHN Distance: 700

================================================

Shortest Route:

================================================
BNG - Bangalore
PUN - Pune
MUM - Mumbai
CHN - Chennai
Distance: 700

================================================

================================================
```

Exiting the System



THANK YOU!

==========