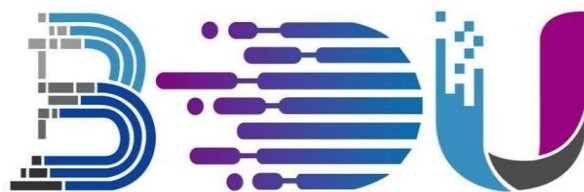


Bangabandhu Sheikh Mujibur Rahman Digital University, Bangladesh



**Bangabandhu Sheikh Mujibur Rahman
Digital University**

Project Report

COURSE CODE - CSE 114

**COURSE TITLE- DATA STRUCTURE AND
ALGORITHMS**

SUBMITTED BY

Abir Kumar Shanto

ID:2201023

Sajid Ahmed Rahi

ID:2201005

Suwaybid Ahmed Muaz

ID:2201020

SUBMITTED TO

Sumon Saha

Lecturer

Department of IRE

Bangabandhu Sheikh Mujibur Rahman

Digital University, Bangladesh

Table of Contents

1. Introduction
2. Objectives
3. System Architecture
4. Data Structures and Algorithms Used
5. Implementation
6. Conclusion
7. Future Enhancements

1. Introduction

The Metro Management System is a sophisticated software solution leveraging Data Structures and Algorithms (DSA) implemented in C++. It aims to optimize metro operations, ensuring efficiency, safety, and passenger convenience.

2. Objectives

Develop a user-friendly interface for seamless interaction.

Implement efficient algorithms for route planning and train scheduling.

Enhance safety measures and optimize resource utilization.

Provide statistical analysis to support management decisions.

3. System Architecture

- **User Interface Module:** Facilitates user interaction for ticket purchasing, route planning, and information retrieval.
- **Ticket Management Module:** Manages ticket transactions, validations, and refunds.
- **Route Planning Module:** Utilizes DSA to calculate optimal routes considering factors like time, distance, and transfers.
- **Train Scheduling Module:** Implements algorithms for efficient train scheduling, considering passenger demand and track availability.

4. Data Structures and Algorithms Used:

Arrays: Arrays will be used to store the stations names.

Graph and Map Data Structure: To represent the network of Metro rail, Graph will be used.

Queues: Priority Ques will be used in Dijkstra's algorithm.

Algorithms:

Dijkstra's Algorithm: Utilize Dijkstra's algorithm to find the shortest path between two stations in the metro network. This algorithm efficiently calculates the shortest distance between nodes in a graph, making it suitable for route planning in metro systems

5. Implementation:

- The system is implemented in C++ for robustness and performance.
- Standard Template Library (STL) is utilized for efficient data structure usage, enhancing code readability and maintainability.
- Real-time scheduling and monitoring functionalities are implemented to ensure accurate and timely information dissemination.

6. Output

```
PS C:\Users\User\Documents> .\a.exe
1.Create account
2.Log in
:-1
Username: sajid
PIN: 1234
1.Create account
2.Log in
:-2
Enter username: sajid
Enter PIN: 1234

Successfully logged in
Press any key to continue . . .
*****

Menu
1.Buy Ticket
2.Available Stations
3.Routes
4.Timing
5.About
6.Exit
:-|
```

```
Menu
1.Buy Ticket
2.Available Stations
3.Routes
4.Timing
5.About
6.Exit
:-2

*****
Station names and its serial number(0 to
0. Motijheel
1. DU
2. Shahbagh
3. Kawran Bazar
4. Farmgate
5. Agargaon
6. Mirpur
7. Pallabi
8. Uttara
*****
```

```

*****
Station names and its serial number(0 to 8)
0. Motijheel
1. DU
2. Shahbagh
3. Kawran Bajar
4. Farmgate
5. Agargaon
6. Mirpur
7. Pallabi
8. Uttara
*****
Route 1: 0 -> 1
Route 2: 1 -> 0 2 7
Route 3: 2 -> 1 3 5 8
Route 4: 3 -> 2 4 5
Route 5: 4 -> 3 5
Route 6: 5 -> 2 3 4 6
Route 7: 6 -> 5 8 7
Route 8: 7 -> 1 6 8
Route 9: 8 -> 2 6 7
*****

```

```

Route 1: 0 -> 1 --> Timing: 8
Route 2: 1 -> 0 2 7 --> Timing: 8.1
Route 3: 2 -> 1 3 5 8 --> Timing: 8.2
Route 4: 3 -> 2 4 5 --> Timing: 8.3
Route 5: 4 -> 3 5 --> Timing: 8.4
Route 6: 5 -> 2 3 4 6 --> Timing: 8.5
Route 7: 6 -> 5 8 7 --> Timing: 9
Route 8: 7 -> 1 6 8 --> Timing: 9.1
Route 9: 8 -> 2 6 7 --> Timing: 9.2

Route 9: 8 -> 2 6 7 --> Timing: 8
Route 8: 7 -> 1 6 8 --> Timing: 8.1
Route 7: 6 -> 5 8 7 --> Timing: 8.2
Route 6: 5 -> 2 3 4 6 --> Timing: 8.3
Route 5: 4 -> 3 5 --> Timing: 8.4
Route 4: 3 -> 2 4 5 --> Timing: 8.5
Route 3: 2 -> 1 3 5 8 --> Timing: 9
Route 2: 1 -> 0 2 7 --> Timing: 9.1
Route 1: 0 -> 1 --> Timing: 9.2

```

```

*****
Menu
1.Buy Ticket
2.Available Stations
3.Routes
4.Timing
5.About
6.Exit
:-1

```

```

*****
Station names and its serial number(0 to 8)
0. Motijheel
1. DU
2. Shahbagh
3. Kawran Bajar
4. Farmgate
5. Agargaon
6. Mirpur
7. Pallabi
8. Uttara

```

```

*****
Route 1: 0 -> 1
Route 2: 1 -> 0 2 7
Route 3: 2 -> 1 3 5 8
Route 4: 3 -> 2 4 5
Route 5: 4 -> 3 5
Route 6: 5 -> 2 3 4 6
Route 7: 6 -> 5 8 7
Route 8: 7 -> 1 6 8
Route 9: 8 -> 2 6 7

```

```

Enter starting: 0
Enter destination: 8

```

```

Total amount to pay: 35

```

```

Shortest path to the destination: 0 -> 1 -> 2 -> 8 -> End
Required time is: 40 Minutes

```

```

Choose payment option:
1.Cash
2.Digital wallet
:-|

```


7. Conclusion

The Metro Management System in C++ effectively utilizes DSA to streamline metro operations, leading to enhanced efficiency and safety. Future enhancements may include IoT integration for track monitoring and machine learning for predictive maintenance, further optimizing system performance.

8. Future Enhancements

- Integration of IoT devices for real-time track monitoring and maintenance.
- Implementation of machine learning algorithms for predictive maintenance and demand forecasting.
- Development of mobile applications for enhanced passenger experience and accessibility.
- Integration of smart card systems for seamless ticketing and payment processing.