Data Structures and Algorithms

Smart MetroTransportation of Chirasmriti

Course Project Report

School of Computer Science and Engineering
2023-24

Contents

1. Course and Team Details

1.1 Course details

Course Name	Data Structures and Algorithms
Course Code	24ECSC205
Semester	III
Division	A
Year	2024-25
Instructor	Mr Mallikarjun Akki

1.2 Team Details

Si. No.	Roll No.	Name
1.	112	Anagha Hegde
2.	115	Suman Mogali
3.	141	Sanmati U.S.
4.	131	Vindhya Hegde

1.3 Report Owner

Roll No.	Name
141	Sanmati U.S.

2. Introduction

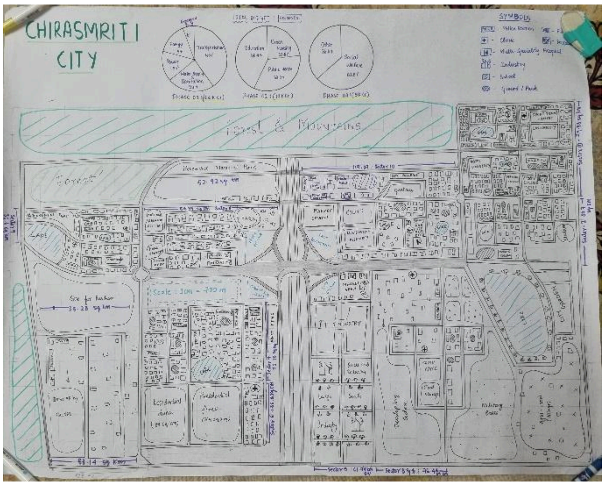
“Chirasmriti: A city of Eternal Memory”

Earlier in this course, a barren land of approximately 1500 sq km was designated for conversion into a smart city. Like other cities around the world, this city will require basic facilities such as infrastructure, healthcare, education, electricity, water systems, and many more. The conceptual design was developed by the team (as shown in the figure). The development of the city is divided into three phases, with the first phase primarily focusing on the establishment of an efficient and sustainable transportation system connecting different parts of the city. This transportation system includes road paving and the development of metro tracks. Hence, the domain of smart metro transportation was chosen.

*What makes metro transportation "**smart**"?*

The integration of advanced technologies to address challenges such as optimizing time scheduling, developing efficient route networks, constructing sustainable railway infrastructure, and effectively managing passenger flow forms the core definition of a smart metro transportation system. However, upon analyzing "The Streets" (white sheet paper), it became clear that these are not the only factors that contribute to making transportation truly smart.The sustainability and stability of the system also contribute significantly to its "smart" nature. The designs introduced must be stable enough to handle future growth. For example, if the population increases exponentially in the coming years, the metro system should remain functional and efficient. The infrastructure must be capable of balancing increased demand. All these considerations were realised after analyzing the white sheet paper.

Through this project, we have aimed to integrate and implement the course topics that are relevant to the development of the city.



3. Problem Statement

3.1 Domain

The first phase of the Chirasmriti City project focuses on the initial development of a smart metro transportation system. The main challenges addressed in this phase under smart metro transportation include the creation of an efficient route networ high passenger volumes during peak hours, ensuring smooth railway infrastructure for quick movement between platforms and issues related to passenger ticket allotment and platform management These problems are explored as separate modules within this domain. These problems were identified based on insights gained from a case study of the local train system in Mumbai, as outlined in the corresponding white paper.

3.2 Module Description

<u>Managing Crowd Density: Designing Smart Solutions for Platform Crowd Management:</u>
Platform crowd management uses real-time data from sensors, cameras, and ticketing systems to monitor passenger movements and crowd density. This helps to predict congestion and adjust train schedules to optimize the passenger flow and enhance safety.

4. Functionality Selection

Si. no.	Functionality Name	Known	Unknown	Principles applicable	Algorithms	Data Structures
	Name the functionality within the module	**What information do you already know about the module? What kind of data you already have? How much of process information is known?**	**What are the pain points? What information needs to be explored and understood? What are challenges?**	**What are the supporting principles and design techniques ?**	**List all the algorithms you will use**	**What are the supporting data structures?**
1	Detecting overcrowded platforms.	(i) Using real-time data, efficient algorithms, and advanced technologies, cities like London(UK), Singapore, Tokyo(Japan) and New York City(USA) predict the congestion, and improve transit efficiency. (ii) TfL(London) uses real-time data to monitor passenger flow across the underground network.	Algorithms that predict crowd behavior and train schedules may rely on historical data that do not accurately predict the current conditions. Integrating multiple systems-train schedules, ticketing, crowd sensors,	(i) **Design Principle:** Rotations for balancing the tree (ii) **Design Principle:** Binary Search Tree (BST) Properties	**AVL tree**	Nodes Balance Factor Parent node pointers

		Advanced algorithms predict crowding, while dynamic signage(digital displays) helps guide passengers to less congested areas. (iii) Singapore's MRT system monitors the crowd density to adjust train schedules and send real-time alerts to about crowded platforms or delays, ensuring smoother operations.	platform monitoring for accurate crowd management.			
2	Rescheduling the train arrivals and departure by prioritizing highly crowded platforms.	The possible algorithms that would help to arrive at the result.		**(i) Design Principle:** Heap Property **(ii) Design Principle:** Array-Based Representation	**Max heap Priority Queue**	Structure of the Heap Node Array

5. Functionality Analysis

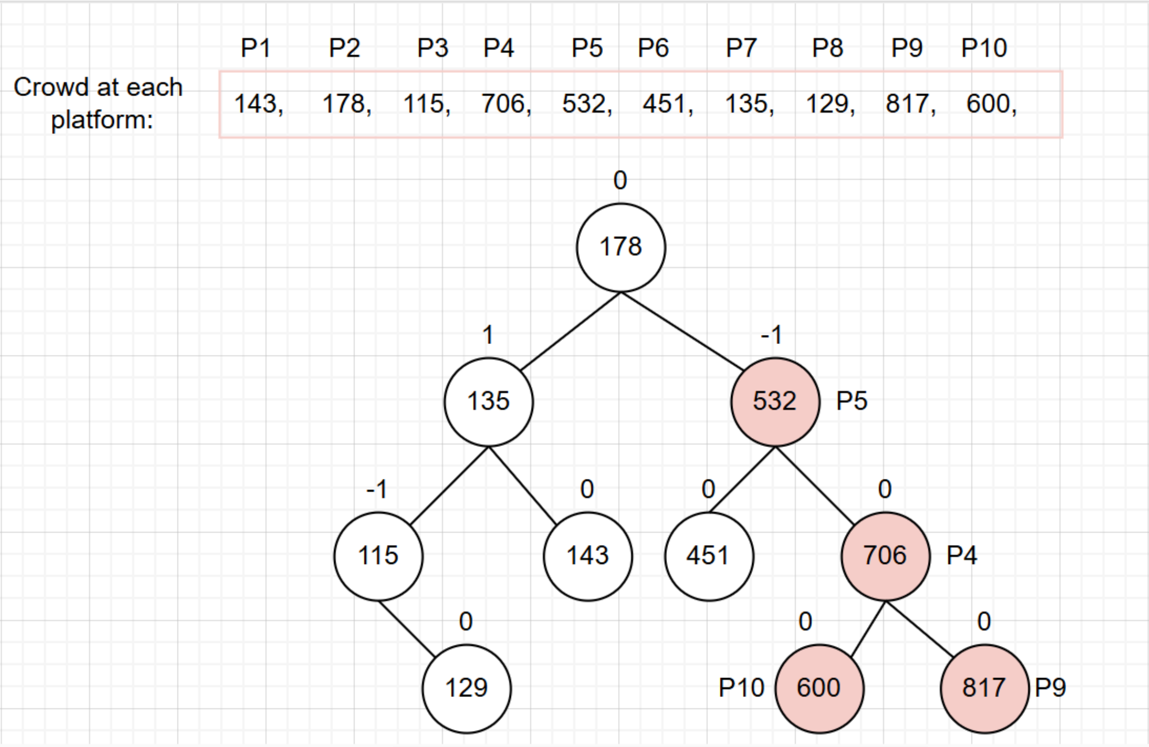
(1)Functionality Analysis of detecting overcrowed platforms:

Assumptions:

- Real-time data is continuously available from platforms through sensc or ticketing systems.
- A **pre-defined crowd density threshold** of **500** passengers exists that determines whether a platform is overcrowded (assume the number of platforms to be **10).**

1. Consider crowd density at each platform at a particular time as the key in the nodes of the **AVL Tree.**
2. As passenger data arrives in real-time (from sensors, cameras or ticketing systems) the **AVL tree** allows dynamic insertion of **platform nodes** with crowd densities.

- Insert each node as per binary search tree properties
- Balance the AVL tree after every insertion.



	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Crowd at each platform:	143,	178,	115,	706,	532,	451,	135,	129,	817,	600,

3. Scan all the platforms to detect the overcrowded platforms (with cr

   500 or more)

- delete those nodes from the AVL Tree. and insert them into priority queue

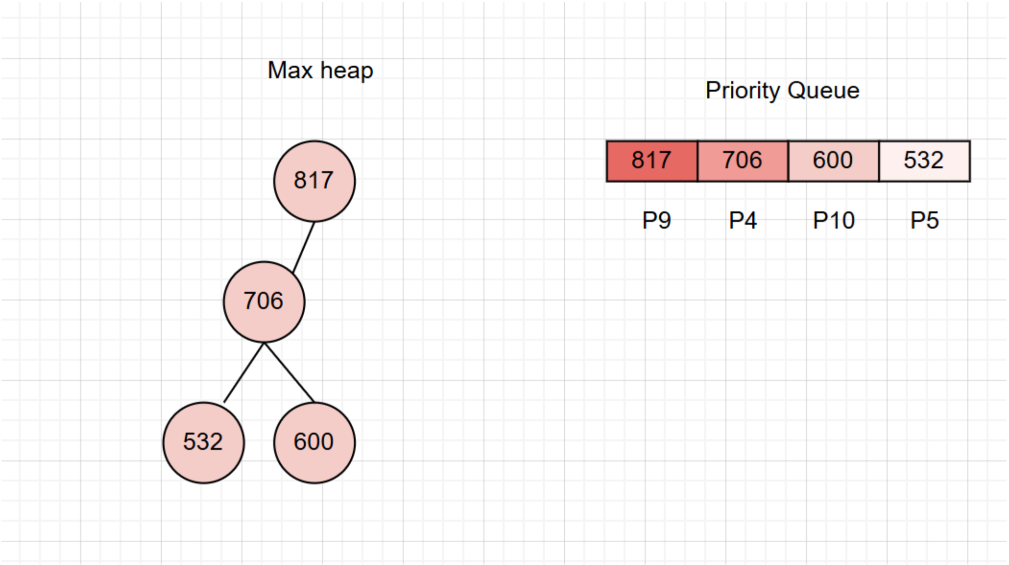
4. Using Max heap priority Queue the platforms are prioritized based on their crowd

   density, ensuring that highly overcrowded platforms are addressed first.

5. The priority queue works alongside the AVL tree to efficiently reschedule trains.

6. Overcrowding in certain platforms can also occur during special events or fetivities .

7. Peak hours can be managed by sending empty trains to the overcrowded platforms.



Time Complexity of an AVL Tree:

->Search: O(log n)

-> Insertion: O(log n)

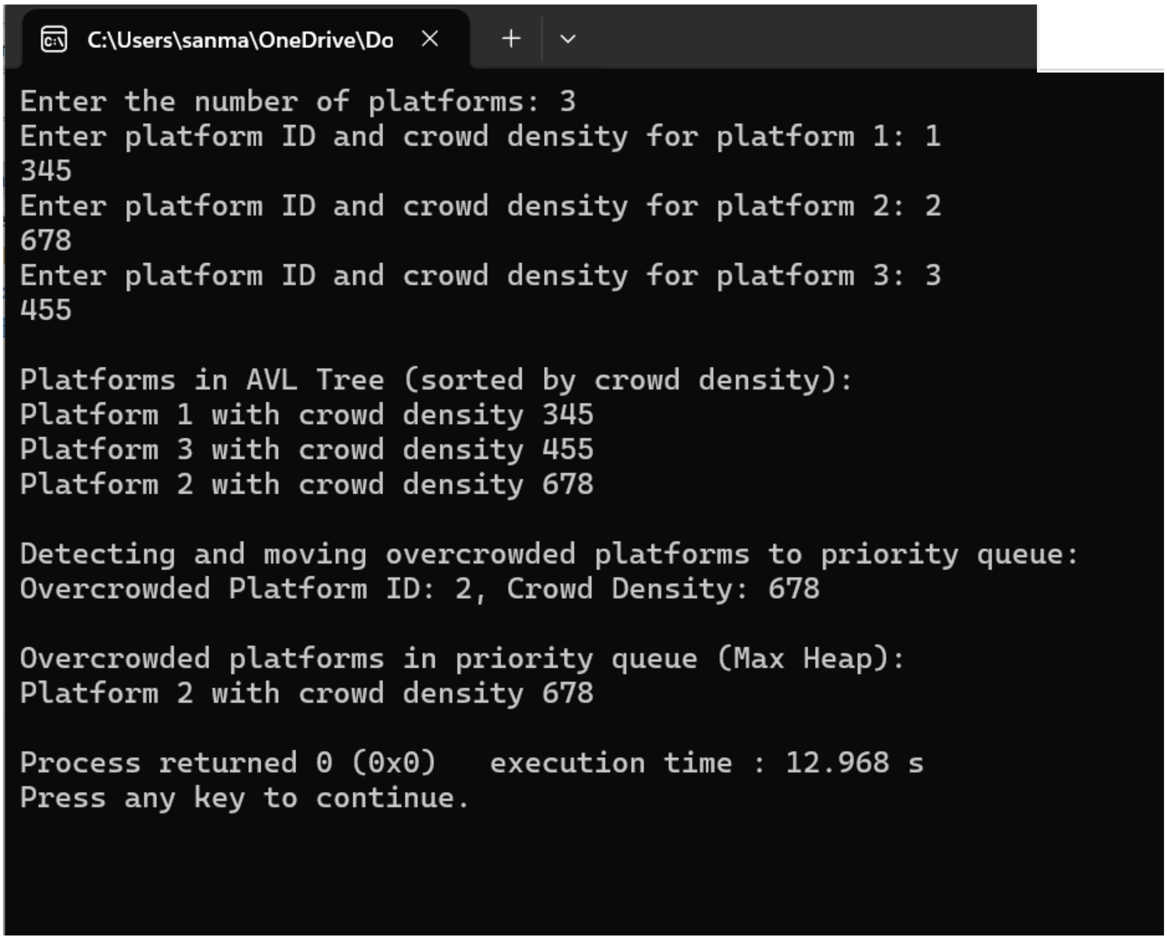
-> Deletion: O(log n)

-> Traversal: O(n)

The picture below is the result after implementing the code using AVL Tree for the information from 3 platforms:

Inputs:
Platform Crowd Density: Real-time number of passengers waiting.

Output:
Platform ID of overcrowded platforms.



```
Enter the number of platforms: 3
Enter platform ID and crowd density for platform 1: 1
345
Enter platform ID and crowd density for platform 2: 2
678
Enter platform ID and crowd density for platform 3: 3
455

Platforms in AVL Tree (sorted by crowd density):
Platform 1 with crowd density 345
Platform 3 with crowd density 455
Platform 2 with crowd density 678

Detecting and moving overcrowded platforms to priority queue:
Overcrowded Platform ID: 2, Crowd Density: 678

Overcrowded platforms in priority queue (Max Heap):
Platform 2 with crowd density 678

Process returned 0 (0x0)   execution time : 12.968 s
Press any key to continue.
```

Conclusion: Based on the result,we can detect the highly crowded platforms which require immediate attention and necessary changes in the train schedules are made for better passenger flow.

Alternate possible algorithms:

1. **Red Black Tree**

   -> Useful for scenarios that require efficient, real-time insertion, deletion, and orcrowd data based on factors like priority, time, density, and urgency.

-> Ideal for systems where quick adjustments are needed to maintai

efficiency.

-> **Time Complexity :** O(log n)

6. Conclusion

- The project encouraged us to think critically about how algorithms can be applied to address real-world challenges and design efficient and smart solutions.
- It helped us to enhance our logical thinking and the ability to think algorithmically when approaching new problems when given a real world scenario.

7. References

- Introduction to the Design and Analysis of Algorithms(3rd Edition) by Michael Levitin
- https://www.reddit.com/r/bangalore/comments/1bo1u7b/bangalore_metro_over_crowding/
- https://www.london.gov.uk/who-we-are/what-london-assembly-does/questions-mayor/find-an-answer/crowd-management-london-underground/

Tools used:

- https://app.diagrams.net/

~*~*~*~*~*~*~*~