Pollution Aware:

A City’s Fight Against Pollution

Larana, Inc.

POLLUTION MONITORING AND ALERT SYSTEM

[CPMAS]

CONTENT

01 Problem Definition

02 Problem Identified

03 Modules

04 Data Structures

05 Algorithms

PROBLEM DEFINITION

- The goal is to develop a City Pollution Monitoring and Alert System (CPMAS) that effectively tracks, analyzes, and forecasts air pollution levels in real time and citizens take proactive measures to combat pollution and minimize health risks.

- Relevance Example Delhi

- Delhi, one of the world’s most polluted cities, faces significant challenges related to air quality management due to factors like vehicular emissions, construction dust, stubble burning, and seasonal changes

PROBLEM IDENTIFIED

- **Lack of Real-Time Data:**
- Many cities lack a real-time pollution monitoring system, making it difficult to track and respond to sudden spikes in air pollution levels.

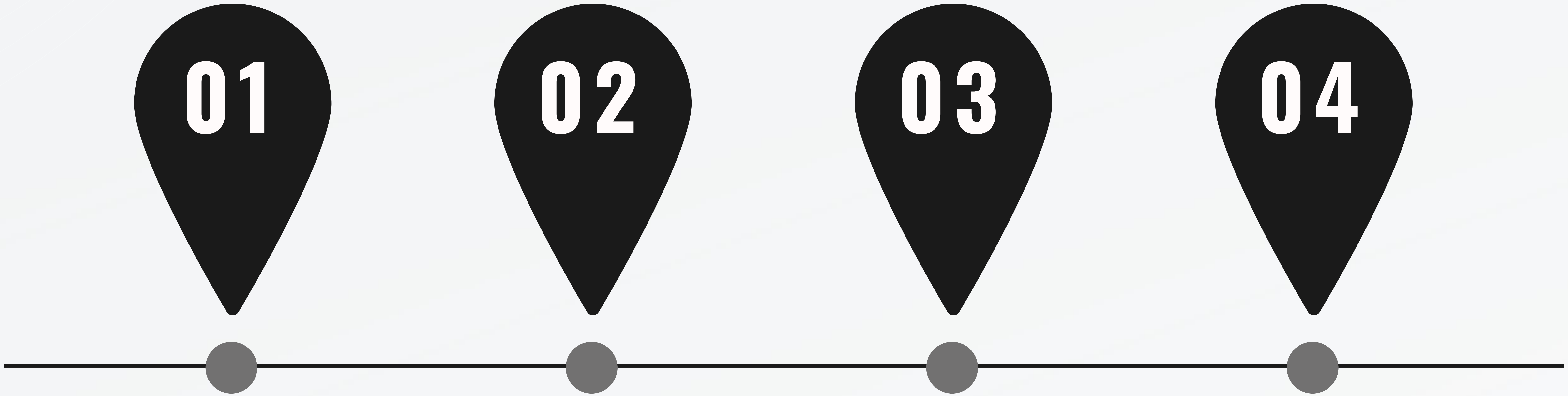
- **Inefficient Threshold Detection:**
- Without automated systems, authorities struggle to identify when pollution exceeds safe limits, delaying necessary interventions or alerts to residents.

- **Limited Geographic Coverage:**
- Existing pollution monitoring systems are often concentrated in specific areas, leaving significant portions of the city unmonitored, especially in underserved or densely populated zones.

- **Ineffective Resource Allocation:**
- Authorities face challenges in allocating resources, such as air purifiers, masks, or emergency medical aid, to the most affected areas due to a lack of actionable data.

- **Health Risks Due to Delayed Alerts:**
- Delayed or inadequate alert mechanisms fail to warn vulnerable populations, such as children, the elderly, and those with pre-existing conditions, leading to severe health consequences.

MODULES



1.Data Collection and Integration

- Collect real-time pollution data from sensors, traffic data, and meteorological inputs.

2.Data Processing and Analysis

- Apply clustering algorithms (e.g., K-Means) to identify pollution hotspots.

3: Visualization and User Interface

- Create a dashboard displaying real-time pollution levels, alerts, and recommendations for mitigation.

4: Alert System and Notifications

- Implement an alert system to notify users about high pollution levels, health risks, and suggested actions.

DATA STRUCTURES

Core Data Structures

Store pollution readings for sensors in a time-series format.

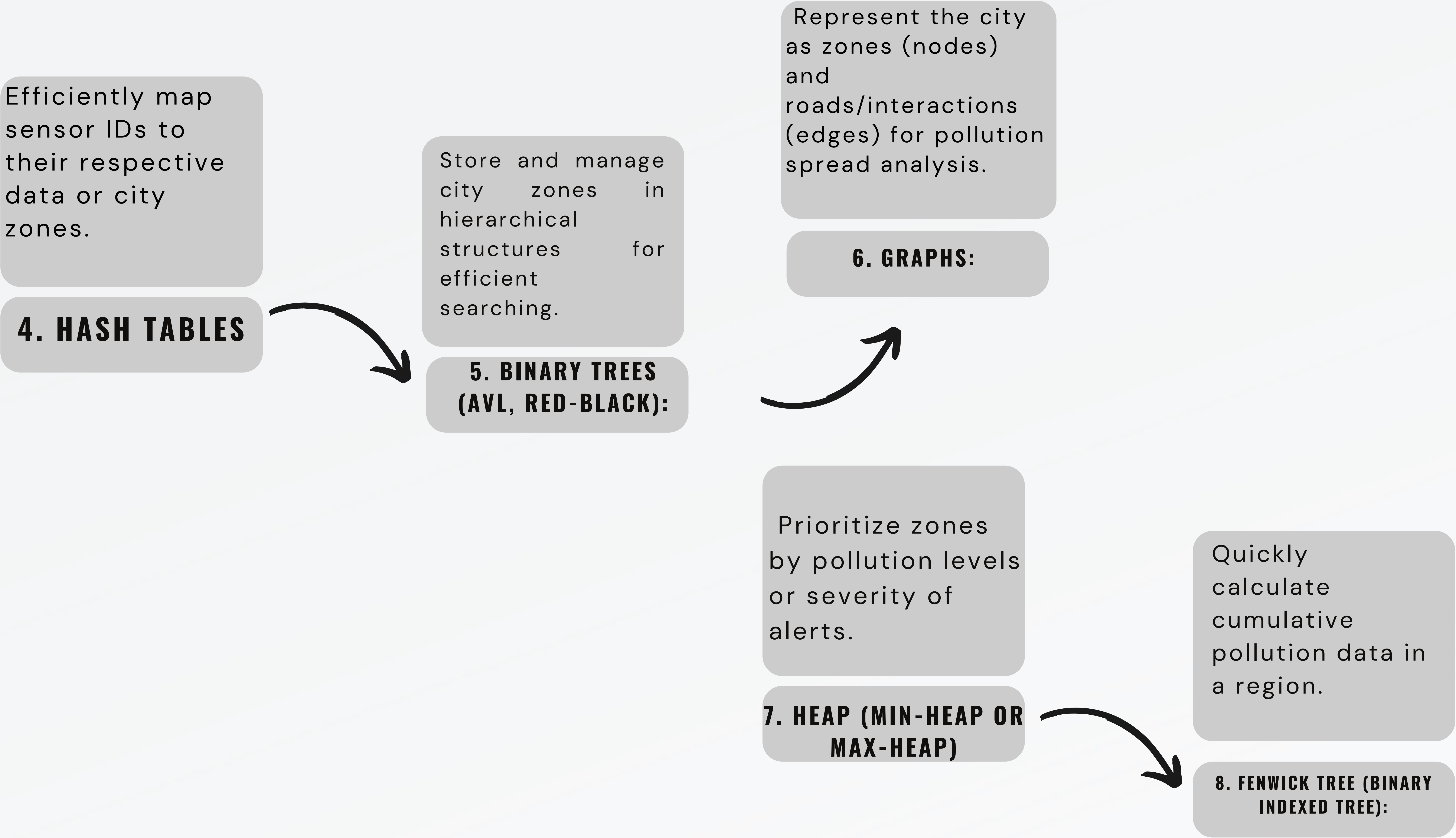
1. ARRAY

Dynamically store historical pollution data for zones

2. LINKED LIST:

Handle alert notifications or user interaction logs.

3. STACKS AND QUEUES



4. HASH TABLES

Efficiently map sensor IDs to their respective data or city zones.

5. BINARY TREES (AVL, RED-BLACK):

Store and manage city zones in hierarchical structures for efficient searching.

6. GRAPHS:

Represent the city as zones (nodes) and roads/interactions (edges) for pollution spread analysis.

7. HEAP (MIN-HEAP OR MAX-HEAP)

Prioritize zones by pollution levels or severity of alerts.

8. FENWICK TREE (BINARY INDEXED TREE):

Quickly calculate cumulative pollution data in a region.

ALGORITHMS

1. Brute Force:
   - Bubble Sort and Selection Sort: For ranking zones based on pollution levels.
   - Linear Search: To query specific pollution data by ID or zone.
   - String Processing: Handle pollution data or alert messages.
2. Divide and Conquer:
   - Binary Search: Quickly identify if pollution exceeds threshold levels.
   - Merge Sort and Quick Sort: Sort zones by pollution levels or other metrics.
3. Decrease and Conquer:
   - Insertion Sort: Maintain sorted lists of pollution records or alerts.

4. Graph Traversal:
   - Breadth-First Search (BFS): Analyze pollution spread across zones.
   - Depth-First Search (DFS): Find connected zones or paths with specific pollution levels.

Additional Algorithms to Include

5. Dijkstra’s Algorithm:
   - Find shortest paths for resource allocation (e.g., air purifiers) in the city.
6. Kruskal’s/Prim’s Algorithm:
   - Build a minimum spanning tree for efficiently connecting zones with pollution control systems

7. Clustering (e.g., K-Means):
   - Group zones with similar pollution levels for targeted action.
8. Sliding Window Algorithm:
   - Calculate moving averages of pollution over time for real-time monitoring.
9. Threshold Detection:
   - Trigger alerts when pollution crosses predefined thresholds (linear or binary search).
10. Dynamic Programming (DP):
    - Allocate resources optimally to minimize overall pollution impact

THANK'S FOR WATCHING

LARANA, INC.