# EXPO WEATHER

FORECASTS WITH EXPONENTIAL TREES

# GROUP MEMBERS

Muhammad Huzaifah Riaz - Front-End
Zarak Khan - Weather API and Integration
Dua E Zehra - Backend/Datastructure
Dilawar Saeed - Backend/Datastructure

EACH GROUP MEMBER EQUALLY CONTRIBUTED TO THE PROJECT I.E. 25%

# INTRODUCTION:

Project Aim

- To develop a GUI-based weather application that utilizes an exponential tree data structure for efficient weather data storage and retrieval.

Key Features

- Real-time weather updates
- 5-day weather forecasting
- Aggregated weather data using an exponential tree
- User-friendly graphical interface

# TECHNICAL ARCHITECTURE

- **System Components:**
  - Frontend: GUI built with Tkinter
  - Backend: Data management using the ExTree data structure
  - API Integration: Real-time data fetched from OpenWeather API

- **Data Flow:**
  - Visual diagram showing the API data flow into the ExTree and then to the GUI.

- **Technologies:**
  - Python for backend logic, Tkinter for frontend

# MOTIVATION

Exponential trees offer an advanced model for applications that deal with rapidly increasing amounts of data or require quick scalability

- *Scalability*
- *Probabilistic Modeling*
- *Efficient Data Handling*
- *Handling of Large Datasets*
- *Quick Insertions and Updates*
- *Dynamic Query Capability*

# WHAT ARE E-TREES?

- Exponential trees store data in a hierarchy, similar to a family tree.
- Each level can have exponentially more nodes than the one above it.
- The top of the tree representing the broadest category of data is the root node.
- Each node may have numerous children, leading to rapid expansion at each level.
- Optimized for quick access to data at various levels of detail.
- Ideal for data with natural hierarchical relationships, such as organizational structures or geographical information.
- While they can store large amounts of data, depth and branching are often limited to maintain efficiency.

# E-TREES IN OUR APP

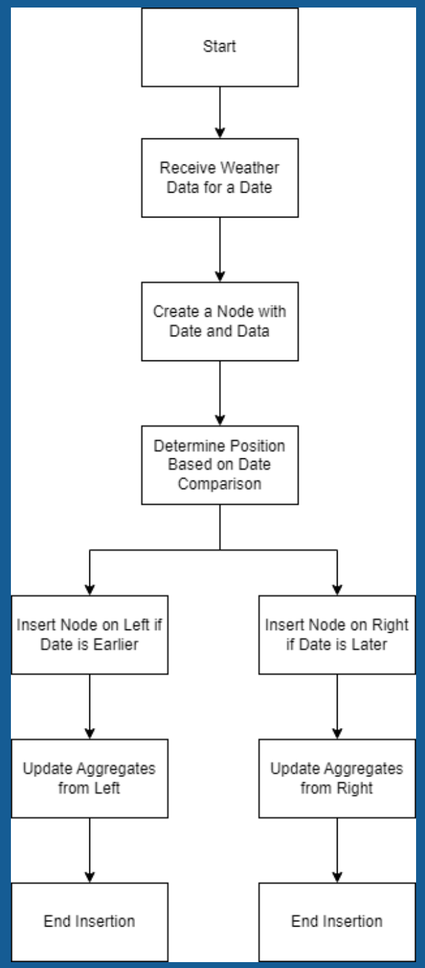
- Structure and Purpose
  - Each node in the ExTree corresponds to a single day's weather data, encapsulated in an ExTreeNode. This node holds not just the weather data for that particular day, but also aggregates from its child nodes.
  - Weather data for each day includes temperature, humidity, pressure, wind speed, and precipitation. This data is stored in each node when new forecast data is inserted into the tree.

- Key Functions and Their Roles
  - **Insertion**: When new forecast data is obtained from the API, it is processed and inserted into the tree.
  - **Process**: Each piece of forecast data is inserted according to the date. The tree organizes itself by dates, ensuring all nodes are sorted chronologically. If a node with the given date already exists, it updates the existing node's data.

**Aggregation**:
  - **Data Retrieval**: This function is used to fetch aggregated weather data over a specific range of dates.
  - **Process**: The function traverses the tree, including only those nodes that fall within the specified date range. As it traverses, it aggregates the data from each node that meets the criteria.
  - **Utilization**: This capability is particularly useful for providing summaries of weather conditions over several days, such as average temperature, total precipitation, etc.
  - **Process**:  which are then formatted and displayed in the application's forecast area.

# FLOWCHARTS

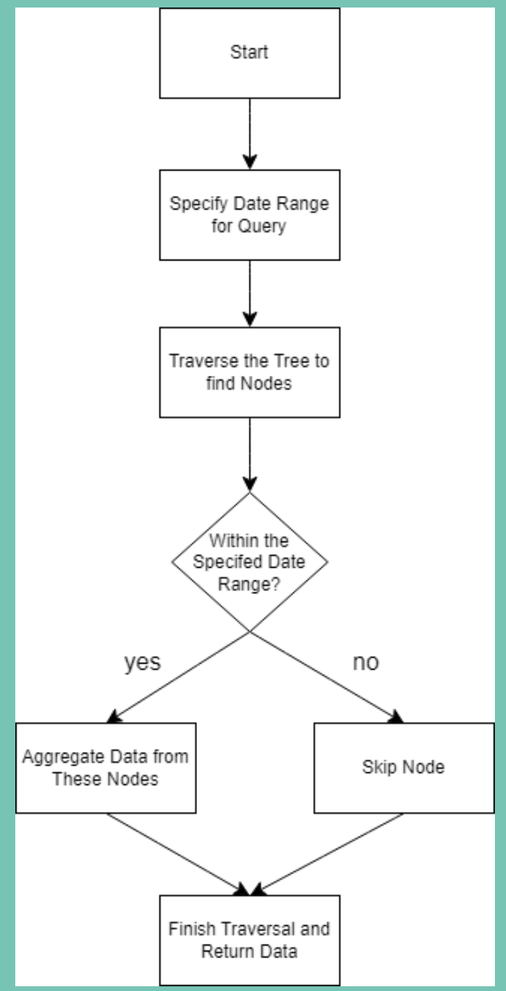
## DATA STORAGE




After inserting a node, the tree updates aggregated information ( sum of temperatures, humidities, pressures, wind speeds, and precipitation from child nodes) both in the new node and in its parent nodes.

## RETRIEVING DATA

After retrieving aggregated data for a specified range, this data is processed to calculate averages and totals



# TIME COMPLEXITY

**insert**
Best Case Olog(n)
Worst Case O(n) or O(2^n)

**update BST**
Best Case O(logn)
Worst Case O(n)

**query range** - depends on query range
Best Case O(logn)
Worst Case O(n)

**avg weather calculation** - heavily depends on query range and inherits its complexity.

## LIMITATIONS

**Complexity in Balancing**
**Implementation Complexity**
**Memory Usage**
**Predictability**

# APP FEATURES

Given the name of any location city in the world our app can output the following information:

**Forecast of next 5 days**

- Avg Temp for next 5 days
- Avg Humidity for next 5 days
- Total Precipiation for next 5 days
- Avg Pressure for next 5 days
- Avg Wind Speed for next 5 days

**Current Weather**

- Temprature
- Humidity
- Precipiation
- Pressure
- Wind Speed

Weather App

Karachi

31.9°

Avg TEMP: 27.86°C
Avg HUMIDITY: 47.75%
Total PRECIPITATION: 0.00 mm
Avg PRESSURE: 1008.75 hPa
Avg WIND SPEED: 6.82 m/s

| WIND | HUMIDITY | PRECIPITATION | PRESSURE |
|---|---|---|---|
| 4.12 m/s | 45.00% | 0.00 mm | 1010.00 hPa |

# CHALLENGES

1. **Limited API Access:** Restricted access to forecast data beyond 5 days due to the limitations of the Open Weather API's free tier.
2. **Lack of Implementation References:** Absence of existing implementations of exponential trees for reference, requiring extensive theoretical understanding and independent development.
3. **Complex Data Structure:** Implementing exponential trees, a relatively complex data structure, without existing code examples or comprehensive documentation.
4. **Data Accuracy and Reliability:** Ensuring the accuracy and reliability of weather data obtained from the API for precise calculations.
5. **Error Handling and Edge Cases**: Implementing robust error handling mechanisms to address unforeseen issues, such as network errors, API downtime, or invalid data responses.
6. **User Interface Design:** Designing an intuitive user interface to present the calculated weather metrics effectively, considering the diverse user preferences and device capabilities.

# THANK YOU !

*REFRENCES:*


- *RAFIK AGUECH, SUDIP BOSE, HOSAM MAHMOUD & YI ZHANG (2022) SOME PROPERTIES OF EXPONENTIAL TREES, INTERNATIONAL JOURNAL OF COMPUTER MATHEMATICS: COMPUTER SYSTEMS THEORY, 7:1, 16-32, DOI: 10.1080/23799927.2021.1974569*

- *FOCS '96: PROCEEDINGS OF THE 37TH ANNUAL SYMPOSIUM ON FOUNDATIONS OF COMPUTER SCIENCEOCTOBER 1996*