

AVL Tree-Based Delivery System

(Archit Mittal – mittalarchit@ufl.edu -58408828)

Project Details :

This report outlines the AVL Tree-Based Delivery System, designed to efficiently manage and process delivery orders using self-balancing binary search trees. The system keeps track of orders with various attributes such as order ID, priority, delivery time, and estimated time of arrival (ETA), providing functionality for adding, updating, and querying orders.

Class Definitions

Class OrderInfo: Represents an individual order with attributes such as order ID, creation time, order value, delivery time, estimated time of arrival (ETA), and priority. It calculates priority based on order value and current system time.

Class AVLNode: A node in the AVL tree, storing key-value pairs where the key is either order priority or ETA, and the value is an OrderInfo object. Each node also tracks its height for AVL balancing.

Class AVLTree: Implements an AVL tree with methods for inserting nodes, deleting nodes, rotating subtrees to maintain balance, and traversing the tree. Two instances of AVLTree manage orders based on priority and ETA, respectively.

Function Prototypes and Explanation:

Global Variables

- **Tree1** and **Tree2:** AVL trees for managing orders based on priority and ETA.
- **sys_time, del_item, last_del_item, del_completed, out_data:** Variables to track system time, the current item to be delivered, the last delivered item, a record of completed deliveries, and output data for reporting

Class OrderInfo

- **__init__(self, orderID, currentSystemTime, orderValue, del_time):** Initializes an order with its ID, creation time, value, and delivery time. It also calculates the order's priority based on its value and the current system time, which influences its delivery sequence.

- **calculate_priority(self, valueWeight=0.3, timeWeight=0.7):** Calculates the priority of an order using a weighted formula. Higher value orders and those entered earlier have higher priorities. This formula ensures a balance between servicing high-value orders promptly and adhering to a first-come, first-served basis.

Class AVLNode

Represents a node in the AVL tree. Each node corresponds to an order, storing the order's priority or ETA as the key and the **OrderInfo** object as the value. It also tracks the left and right child nodes and its height in the tree, facilitating AVL balancing operations.

Class AVLTree

Implements an AVL tree that maintains balance after insertions and deletions to ensure that operations are performed efficiently.

- **insert(self, node, key, value):** Recursively inserts a new node into the tree based on the key, maintaining the AVL property. After insertion, it checks and corrects the tree balance using rotations.
- **delete(self, node, key, orderID):** Recursively finds and deletes a node with the given key and order ID. If duplicate keys exist, it specifically targets the node with the matching order ID. Balancing is performed after deletion to maintain the AVL property.
- **rotateLeft(self, node)** and **rotateRight(self, node):** Perform left and right rotations on nodes to rebalance the tree. Rotations are used when the tree becomes unbalanced due to insertions or deletions.
- **rebalance(self, node):** Checks the balance of a node and performs appropriate rotations to rebalance the tree.

Global Functions

- **eta_calculation(received_order):** Calculates the ETA for a new order. It places the order in the sequence based on its priority, adjusts the ETAs of subsequent orders if necessary, and inserts the order into both AVL trees (Tree1 for priority and Tree2 for ETA).
- **deliverAfterCurrentSysTime():** Checks if the order currently designated for delivery has reached its ETA. If so, it marks the order as delivered, removes it from the AVL trees, and selects the next order for delivery based on priority.

- **deliverAfterQuit():** When the system is quitting, this function ensures all remaining orders are marked as delivered in their sequential order based on ETA.
- **deliverBeforeCurrentTime():** Identifies and delivers all orders whose ETAs are before the current system time, updating the AVL trees accordingly.
- **createOrder(orderID, currentSystemTime, orderValue, del_time):** Handles the creation of a new order. It calculates the order's priority and ETA, inserts it into the AVL trees, and checks for any orders that need to be delivered immediately.
- **cancelOrder(orderID, currentSystemTime):** Cancels an undelivered order. It removes the order from the AVL trees and adjusts the ETAs of subsequent orders as needed.
- **updateTime(orderID, currentSystemTime, new_del_time):** Updates the delivery time for an existing order. This involves recalculating the order's ETA, removing and reinserting it into the AVL trees, and adjusting the ETAs of subsequent orders.

Logic and Approach

The script uses two AVL trees to manage orders efficiently:

- **Priority-Based Tree (Tree1):** Manages orders based on their priority, which considers both the order value and the time of entry. This tree helps determine the sequence of deliveries.
- **ETA-Based Tree (Tree2):** Organizes orders by their estimated time of arrival (ETA), ensuring that deliveries are made in a timely manner.