For Batch – R2

Smart Car Parking System Using Trees

You are assigned with a task of developing an application for the smart car parking lot system. The car parking lot has 50 parking spaces with parking space ID ranging from 1 to 50 and it can accommodate 50 cars at a time. The status of each parking space is either 0 or 1. 0 indicates that the parking space is free. 1 indicates that the parking space is occupied. The parking lot provides discounts and additional benefits if the users (vehicle owners) have any membership.

Note: Use B tree, B+ tree or AVL tree to maintain all database.

Hint: You may need to store entries for parked vehicles and status of parking spaces separately.

Every time a car enters the parking lot; the following details need to be stored in the database. (This is done only if the parking lot has available free parking spaces.)

1. If the vehicle is entering the parking lot for the first time, the details of the vehicle should be registered to the database with the following parameters.

   i) Vehicle number (Which is a unique number)

   ii) Name of the vehicle owner

   iii) Date/Time of arrival of the vehicle for parking

   iv) Date/Time of departure of the vehicle

   iv) Membership if any (Since the vehicle is a newly registered one, there will not be any membership).

   v) Total parking hours: Number of hours the vehicle was parked in the past. (The value will be zero for newly registered vehicles)

   vi) Parking space ID allotted to the vehicle according to the **Allocation policy**.

2. If the vehicle is already registered, search for the vehicle in the database using its vehicle number and update the time of arrival of the vehicle for parking. Update the status of parking space that is allocated as occupied.
3. While exiting the following operations need to be performed

   i. Number of hours the vehicle has been parked should be calculated.

   ii. Add these hours to the total parking hours.

   iii. The membership of the user should be updated according to the **Membership policy.**

   iv. The vehicle should be charged according to the parking lot's **Payment policy**.

   v. Change the value of parking space from occupied to free.

4. Arrange the tree of vehicles based on number of parkings done.
5. Arrange the tree of vehicles based on parking amount paid. Print all the vehicles within two parking amounts given by the user.
6. Arrange the tree of parking spaces based on their occupancy. The top parking space is the one which is occupied most often.
7. Arrange the tree of parking spaces which generated maximum revenue.

Allocation Policy:

i) A nearer parking space should be allocated for the golden and premium membership users. Parking space 1 is the nearest and parking space 100 is the farthest. You can assume that parking spaces 1 to 10 are reserved for golden membership whereas parking spaces 11 to 20 are reserved for premium memberships.

ii) The users with no membership may be allocated a parking space beyond 20. The first available parking space beyond 20 is allocated for a new vehicle entering.

Membership Policy:

i) A Premium membership should be given to the users whose total parking hours reach 100 hours.

ii) A Golden membership should be given to the users whose total parking hours reach 200 hours.

iii) No membership for the users whose total parking hours is less than 100 hours.

Payment Policy:

i) 100 Rs for the first 3 hours and 50 Rs for every extra hour the car was parked.

ii) A discount of 10% should be given if the user (vehicle owner) has membership.

Note:

- Generate initial data for at least 20 parking spaces (gold, silver, and without membership) using file handling.
- Avoid use of arrays.
- Implement above functions using appropriate tree structure. (B tree, B+ tree or AVL tree to maintain all database)