Runway Reservation System

Introduction

Suppose you are a developer in a software company which has got the contract to develop an airport runway reservation system. Currently you are working on a system that has only one runway as part of the airport. The reservation system is going to work over a 24 hour time format.

Your system should allow the reservations for half an hour intervals, if the runway is available for requested time slots for landing or take off.

Housekeeping points

- This is a minimal example and may not follow some standard practices.
- We focus on the main flow, and not much error handling.

Program Organization

The simple program is structured in various layers.

1. **runway_reservation_system.py:** In this python file there are two classes designed. Class Node is used to create the tree nodes for the BST. Other class is created named as runway_bst that will have implementation of different methods such as insert, make_reservation, runway_busy and validate interval. These methods are designed to perform specific operations. The implementation for these methods and related information is mentioned in the next section.

Problem Statement

In the given python file you have to implement the incomplete methods. These methods and their behavior is mentioned following:

1. (**Mandatory**) Add code for make_reservation()
    a. You have to implement the above mentioned method. This method is supposed to be called from the driver code. This method will insert the data after validating two operations.
        i. make_reservation: Each plane can book the available slot of the runway for only 30 minutes. If any request comes for more or less than that then the request will be denied. For the error message look at the **runway_system.txt.** This method will also check if the runway is free

during the asked time by calling the runway_busy method. After verifying that runway is available you can perform the insert operation.

ii. runway_busy: This method is supposed to check if the runway is available for booking in the given duration and there is no overlapping. This condition should be checked only after validating the interval.

iii. insert: This is the underlying method behind the reservation making method. The behavior of insert method is similar to the BST insert operation where by comparing the key with the root value, the ideal position for the node will be identified. After that the new node will be inserted in that position.

Evaluation Rubric

Total Project Points: **100**

- Basic compilation without errors (**10%**) : **10 Points**
- Correctness:
  Correctness of implementation
    - Problem statement - point 1.a.(i) (**30%**) : **30 Points**
    - Problem statement - point 1.a.(ii) (**30%**) : **30 Points**
    - Problem statement - point 1.a.(iii) (**30%**) : **30 Points**

Program Instructions

1. Download the zipped folder named **M02-W07-08-Reservation-System.zip**, unzip it on your local machine, and save it. Go into the directory named **M02-W07-08-Reservation-System.**

2. Make sure you have Python 3.6 or higher installed. At your command prompt, run:

   ```
   $ python --version
   Python 3.7.3
   ```
   If not installed, install the latest available version of Python 3.

3. Open the unzipped folder, and make sure that you have some code available in the added file. Write the code based on the given task, you can run the python file to verify if the added code is working successfully or not.

   ```
   $ python3 runway_reservation_system.py(On many Linux platforms)
                               OR
   $ python runway_reservation_system.py   (On Windows platforms)
   ```
   In any case, one of these two commands should work.

4. After running **runway_reservation_system.py,** you can examine the result. This file will currently run various simple calls that will communicate with different classes and methods in those classes. As you solve the problems, you'll be frequently modifying and running this file. You can comment or modify the initial code as needed. To verify, if the accuracy of your program look at the attached runway_system.text file.

5. Alternatively, you could install a popular Python **IDE**, such as **PyCharm** or **Visual Studio Code**, and select a command to build the project from there.A helper file/document that has necessary information about how to use PyCharm for the first time will also be available.

6. There will be another file that will help you to understand the import mechanism in python will also be available.Please take help of these files accordingly.

7. Once the program is ready to submit, zip the parent folder **M02-W07-08-Reservation-System**, and upload the new zip file as **M02-W07-08-Reservation-System.zip**. It is now ready for evaluation.