

COMP 200
Data Structures and Algorithms
Programming Assignment # 4

You are tasked with building an automatic appointment reservation system for a very busy heart surgeon in the city. The system is going to work as follows:

a. The patient requests for an appointment time. The system looks through the list of appointments and if there are no appointments scheduled within 10 minutes either way of the requested time, then that appointment is added to the list of appointments along with the patient ID. Otherwise, the patient is given the next available time automatically that is not within 10 minutes of any other scheduled appointment.

b. At the appointed time, the appointment is marked as serviced and deleted from the list of appointments.

You must use an AVL Tree to implement this appointment system. The reservation system should run as follows:

1. It should run for a total of $12 \times 25 \times 60$ minutes.
2. At each time instant, there is a 10% probability that a new request comes in. If a new request comes in, generate a random number between the current time and (current time + 1000) for the value of the patient request time. A unique patient ID is also generated and the (appointment time, patient ID) is added as a (key, value) pair to the AVL Tree.
3. To do the 10 minute check, what you can do is that during insertion, on the path from the root while looking for a place to insert the new node, if any node is encountered that is within 10 minutes of the desired time, then you know that the 10 minute check has failed. In that case, use the `find_range(...)` function of your AVL Tree class to automatically find the next available time and add it to the AVL Tree.
4. After every 100 time instances, display the number of appointments scheduled in the next 100 minutes.
5. At the end of the simulation, display the average time between a patient's requested time and the time he ended up being allotted.
6. Also display the number of patients who were serviced and how many appointments are still in the AVL Tree at the end of the simulation.