CS234 Fall 2017 Assignment 1

Due at 4:00 PM on Friday, September 29

Submit using MarkUs

Note: For all programming questions, you must use Python 3.2.3 or higher. Consider using the design recipe and Python style guide available from the course website.

Programming Component

1. An appointment book (for a single year) is used for booking appointments. Suppose appointments can only be made at full or half clock between 8AM and 5PM every day. When making appointments, we need to record date, time and purpose of the appointment. Given the following list of operations, implement **AppointmentBook ADT**. (Note: To simplify the implementation, we represent dates using an integer number between 1 and 365; time using a float number, e.g. 8.0 for 8AM, 13.5 for 1:30PM.)

    - *AppointmentBook():* creates an empty appointment book. (5 marks)
    - *isAppointment(apptDate, apptTime):* determines if an appointment exists for the date and time specified. (5 marks)
    - *makeAppointment(apptDate, apptTime, purpose):* inserts the appointment for the date, time and purpose specified as long as it does not conflict with an existing appointment. Returns True if successful, False otherwise. (10 marks)
    - *cancelAppointment(apptDate, apptTime):* deletes the appointment for the date and time specified. Returns True if successful, False otherwise. (10 marks)
    - *checkAppointment(apptDate, apptTime)*: retrieves the purpose of the appointment at the given date and time, if one exists. Otherwise, returns a Null string. (10 marks)
    - *changeAppointment(oldDate, oldTime, newDate, newTime):* change the date or time for an appointment. Returns True if successful, False otherwise. Print information on the screen such as: “You already have an appointment at … on … “ or “Your appointment has been rescheduled to … on …” or “You do not have an appointment at … on …”. (10 marks)
    - *getAppointmentsByDate(date):* retrieves all the appointments on the given date. Returns a list of time and purpose. An empty list returns if no appointment on the date. (10 marks

2. (10 marks) Write a function *buildApptBook*, which reads in a text file *appointments.txt* and uses the AppintmentBook ADT. The file will include each appointment record in a line. Eg.

Make	45	10.5	meeting		
Cancel	80	14.5			
Change		80	15	85	15
Make	120	12	discussion		
……					

3. (10 marks) Write a function *busiestDate*, using AppointmentBook ADT. It returns the date(s) in the list, which includes the date(s) that has the largest number of appointments.

Writing Component

4. Suppose that we are given an array *A* of n > 0 integers such that each of them is in the range *0, 1, …, k* for some positive integer $k \in O(n)$. We want to determine if there exist an integer that appears more than once in *A*.
   - Design and analyze an algorithm that runs in *O(n log n)* worst-case time. (10 marks)
   - Design and analyze a better algorithm that runs in *O(n)* worst-case time. (10 marks)

   For each algorithm, you may use an *O(n)* space in addition to input array *A*.