CSE 2010, HW5, Fall 2018

Due Wed Oct 31 at the start of your lab section; Submit Server: class = cse2010, assignment = hw5S*x*Individual

Due Wed Oct 31 at the end of your lab section; Submit Server: class = cse2010, assignment = hw5S*x*GroupHelp

x is 2, 3, or 4—your section number (or j for java submissions).

To improve the experience of customers, an online social network site would like to add a feature that can display and share a timeline of major events/photos of a user. The user might have many events, given a time range, how would you find/display events efficiently?

The goal of HW5 is to manage the timeline and allow the user to specify a time range to display/share his/her major events. To improve efficency, your implmementation uses a skip list that includes the following operations:

- get(*skiplist*, *key*) // if *key* exists, return value associated with *key*; otherwise, return NULL
- put(*skiplist*, *key*, *value*) // if *key* doesn't exist, add entry and return NULL; otherwise, replace value and return the old value
- remove(*skiplist*, *key*) // if *key* exists, remove entry and return its value; otherwise, return NULL
- ceilingEntry(*skiplist*, *key*) // return the entry with the smallest *key* greater than or equal to key; return null if no such entry exists
- floorEntry(*skiplist*, *key*) // return the entry with the largest *key* less than or equal to key; return null if no such entry exists
- subMap(*skiplist*, *key*1, *key*2) // return all entries with *key* such that $key1 \leq key \leq key2$

Use getRandHeight() in fakeRandHeight.c (FakeRandomHeight in java) for put(*key*, *value*) (to facilitate eaiser debugging and testing) [gcc -o hw5 hw5.c fakeRandHeight.c]. Program files are on the course website. We will be evaluating your submission on code01.fit.edu; we recommend you to ensure that your submission runs on code01.fit.edu.

Input: Input is from the command-line arguments for hw5.c:

- filename of actions, each line has one of the following actions:
  - DisplayEvent *date*
  - AddEvent *date event*
  - DeleteEvent *date*
  - DisplayEventsBetweenDates *startDate endDate*
  - DisplayEventsFromStartDate *startDate*
  - DisplayEventsToEndDate *endDate*
  - DisplayAllEvents
  - PrintSkipList

For simplicity, dates are in MMDD format (MM is 01-12, DD is 01-31). You may assume the dates are unique (a more detailed timestamp would be unique in the real-world) and each event is at most 100 characters. You may assume the skip list can have a height of at most 10. Sample input is on the course website.

Output: Output goes to the standard output (screen), each line has a result for the corresponding action:

- DisplayEvent *date event/none*
- AddEvent *date event success/replacingExistingEvent*
- DeleteEvent *date success/noDateError*
- DisplayEventsBetweenDates *startDate endDate date*1:*event*1 ... or *none*
- DisplayEventsFromStartDate *startDate date*1:*event*1 ... or *none*
- DisplayEventsToEndDate *endDate date*1:*event*1 ... or *none*
- DisplayAllEvents *date*1:*event*1 ... or *none*
- PrintSkipList
  (S*h*) empty
  ...
  (S1) *date*1:*event*1 ...
  (S0) *date*1:*event*1 ...

Sample output is on the course website.

Submission: Submit hw5.c that has the main method, and other program files. Submissions for Individual and GroupHelp have the same guidelines as HW1.

Note the late penalty on the syllabus if you submit after the due date and time as specified at the top of the assignment.