

CSE 2010, HW5

Due Tue Nov 5 at the start of your lab section; Submit Server: class = cse2010, assignment = hw5SxIndividual

Due Tue Nov 5 at the end of your lab section; Submit Server: class = cse2010, assignment = hw5SxGroupHelp
 x is 2 or 3—your section number (or j for java).

Consider a digital personal assistant similar to Alexa and Siri that could manage your schedule. How would you design a tool such that it can efficiently find events during a certain period of time (without the speech recognition/generation component)?

The goal of HW5 is to allow the user to specify a time range to get the corresponding events. Also, we would like the user to be able to add and cancel events as the user's schedule changes. To improve efficiency, your implementation uses a Skip List that has at least the following operations:

- $\text{get}(k)$ [handout: p. 403; pseudocode is on p. 439]
- $\text{put}(k, v)$ [p. 403; pseudocode is on p. 440]
- $\text{remove}(k)$ [p. 403; p. 441]
- $\text{subMap}(k_1, k_2)$ [p. 428]

Use $\text{getRandHeight}()$ in `fakeRandHeight.c` (`FakeRandomHeight.java` in java) for $\text{put}(\text{key}, \text{value})$ (to facilitate easier debugging and testing) [`gcc -o hw5 hw5.c fakeRandHeight.c`]. You may rewrite/modify `doublyLinkedList.c/h` (`DoublyLinkedList.java`). Program files are on the course website. We will be evaluating your submission on `code01.fit.edu`; we strongly recommend you to ensure that your submission runs on `code01.fit.edu`.

Input: The command-line argument for `hw5.c` (`HW5.java`) is the name of a file, which has one of the following actions on each line:

- `AddEvent time event`
- `CancelEvent time`
- `GetEvent time`
- `GetEventsBetweenTimes startTime endTime`
- `GetEventsForOneDay date`
- `GetEventsForTheRestOfTheDay currentTime`
- `GetEventsFromEarlierInTheDay currentTime`
- `PrintSkipList`

For simplicity, time is an integer in MMDDHH format and date is in MMDD format (MM is 01-12, DD is 01-31, and HH is 00-23) [leading zeros are optional]. The timestamps are unique and each timestamp can only have at most one event. Sample input is on the course website.

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

- `AddEvent time event` [`ExistingEventError:existingEvent`]
- `CancelEvent time event/NoEventError`
- `GetEvent time event/none`
- `GetEventsBetweenTimes startTime endTime time1:event1 ...` or `none`
- `GetEventsForOneDay date time1:event1 ...` or `none`
- `GetEventsForTheRestOfTheDay currentTime time1:event1 ...` or `none`
- `GetEventsFromEarlierInTheDay currentTime time1:event1 ...` or `none`
- `PrintSkipList`
(Sh) empty
...
(S1) `time1:event1 ...`
(S0) `time1:event1 ...`

Sample output is on the course website.

Submission: Submit `hw5.c` (`HW5.java`) 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.