CS 6301.001. Implementation of data structures and algorithms
Long Project 3: Skip Lists

Ver 1.0: Initial description (Thursday, Oct 10).
Due: 11:59 PM, Sun, Oct 27 (1st deadline), Sun, Nov 3 (2nd deadline).

Max excellence credits: 1.0.

- **Submit before the first deadline to be eligible for excellence credit.**
- Submission procedure is same as the same as that of prior projects.
- For each group, only its last submission is kept and earlier submissions are discarded.
- Your code must be of good quality, well commented, and pass all test cases within time limits to earn excellence credits.

Project Description

Implement the following operations of skip lists. Starter code is provided. Do not change the signatures of methods declared to be public. You can add additional fields, nested classes, and methods as needed. Driver code is also provided along with the several testcases.

`add(x)`: Add a new element x to the list. If x already exists in the skip list, replace it and return false. Otherwise, insert x into the skip list and return true.

`ceiling(x)`: Find smallest element that is greater or equal to x.

`contains(x)`: Does list contain x?

`first()`: Return first element of list.

`floor(x)`: Find largest element that is less than or equal to x.

`get(n)`: Return element at index n of list. First element is at index 0. Call either getLinear or getLog.

`getLinear(n)`: O(n) algorithm for get(n).

`getLog(n)`: O(log n) expected time algorithm for get(n). **This method is optional, but code it correctly to earn EC.**

`isEmpty()`: Is the list empty?

`iterator()`: Iterator for going through the elements of list in sorted order.

last(): Return last element of list.

rebuild(): Reorganize the elements of the list into a perfect skip list. **This method is optional, but code it correctly to earn EC.**

remove(x): Remove x from the list. If successful, removed element is returned. Otherwise, return null.

size(): Return the number of elements in the list.