Lab 3 Skip List – Readme File

File Includes Within This project:

Main.java Event.java Eventlist.java EventReader.java Query.java QueryReader.java Tokenizer.java autograder.txt JRE System Library

Algorithms & Project Issue

Use Skip List data structure to store events, and implement insertion, deletion, finding most recent years and finding range methods.

Part one

In the Event class, instead of using ArrayList, I add three instance variables , *Event[] next*, *Event[] prev* and *Event link*, to construct a node (event object) with pillar and chaining. *Event[] next* and *Event[] prev* are used to point nodes on the right or left of a certain node. *Event link* is used to link nodes with same year in a chaining.

In the Eventlist class.
First, initialize the skip list, creating a head and a tail, as well as fixing the height of them to be 2000. So, I don't need to worry that the height of other nodes could exceed the head or tail.
I create three extra methods, findYear(int year), findLatestYear(int year) and findRangeHelper(int first, int last) to help me realize the implementations.

findYear(int year) takes in a given year, and returns a node with the year if the node exists, otherwise return null.

findLatestYear(int year) takes in a given year, and returns a node with year latest to the given year if the node is not the head, otherwise return null;

findRangeHelper(int first, int last) takes in an interval, and returns the number of nodes in the interval. If there are not any nodes in the interval, it returns 0;

In the insert() method, firstly, call findYear(int year) method to detect if there have already existed a node whose year is the same with the the given year. If there is, link the node which is to be inserted behind the existed node with same year; if there isn't, insert the node into the skip list directly.

In the remove(int year) method, firstly, call findYear(int year) to locate the node with the given year in the skip list. Secondly, remove it from the skip list. Finally, clear nodes with same year in the chaining.

In the findMostRecent(int year) method.
Firstly, call findYear(int year) method to find a node with the given year. If it doesn't exist, call findLatestYear(int year) method to find the latest year <= the given year, and make a pointer y

point the node. Next, call findRangeHelper(y.year, y.year) to get the number of nodes with the given year. Finally, make a pointer k traverse the chaining and push nodes in the chain into an Event array called *latest*.

In the findRange(int first, int last) method.
Firstly, call findRangeHelper(int first, int last) to get the number of nodes in the interval. If there are no nodes in the interval, return 0;
Next, call findYear(first) to locate a node with the year *first* in the left most of the interval. If it doesn't exist, call findLatestYear(first) to find the node with the latest year < first, and make a pointer x point it. Then, make x point x.next[0] which is the node on the left most in the interval.
Second, do a simple list walk to find the rest of the range.
Finally, make a pointer k traverse each chain and put each node within the range into an Event array named range.

Part two
(a) Instead of fixing head's height, dynamically double it when inserting a new node with height exceeding the head's height.

Since, after doubling the head's height the original information in the head's pillar will disappear, I create a method "repoint()", which takes in doubly-sized array and the original array (copy) of the head. The method relink the head's pillar to other nodes according to the data stored in the "copy".

(b) Rather than a doubly linked skip list, in this part, the skip list becomes a singly linked one. Which is, every node in the skip list contains only next pointers.
When linking a node e between node x and node y, x.next[l] = e; e.next[l] = y; works.
When removing a node e in the skip list, firstly searching for a node h1, whose next[l] pointer points e. Then do h1.next[l] = e.next[l];

Issues
At the beginning, the cost for findRange() is proportional to the number of years between first and last. This is because I use "year = year - 1" to find first and last.
After asking professor Buhler for help, I implemented by finding the latest year <= first and do a list walk to find the rest of the range.