

Cs352 — Homework #5

SkipList and Persistent Data Structures

(Version 2)

October 9, 2003

Due Time: 10/23/03 (9:00PM). Submission in pairs is allowed and encouraged.

Turnin ID: cs352_assg5

Turnin File List: “Makefile” & whatever number of .c or .h files with whatever filenames

In this exercise you have to implement the two-dimensional data structures for processing a set of horizontal segments.

1 General Instructions

1. No two segments in the input have the same y -coordinate.
2. No two endpoints have the same x -coordinate.
3. Each file that has been opened, has to be closed.
4. The input is legal. There is no need for validity checks.
5. All files should be in the current working directory.
6. You should check if opening of the input or output file fails, and if so print an error message and `exit(1)`.
7. Your data structure should answer each query in time $O(\log n)$. Insertion and deletions of items as well as memory used should be as efficient as possible — as follows:
 - (a) To obtain 100% of the possible grade, it is enough to generate every people SkipList (see the slides of the Persistent Data structure for terminology)

from the previous people SkipList by copying the whole people SkipList, and performing insert/delete on the newer copy.

- (b) To obtain 10% bonus points, you should perform deletion in the people SkipList by copying the previous people SkipList. Insertion however should be accomplished using the “virtual copy” method studied in class, and change the deletion function accordingly.
- (c) To obtain 25% Extra bonus points, both insertion and deletion should be accomplished by the virtual copy methods studied in class (note that you need to use the function **follower** from the slides.

- 8. No query point has the same x -coordinate of the endpoint of a segment.
- 9. Points will be taken off for insufficient code comments.
- 10. Make sure to submit a proper “Makefile” file. For command “make” it should generate an executable named “PSkipList” in the current working directory. With the aid of this “Makefile”, you can submit whatever number of files with whatever filenames, as long as the “make” command can generate the expected executable named “PSkipList”. During the grading, we will just run a “make” command then run the executable “PSkipList” generated. Be sure to use -Wall option in your gcc invocation.

Your program should read an input file called “input_segments.inp” (format below). The program has then to create the data structure needed. Then it should read a file called “queries.inp”. Each line in this file contains the coordinates of a query point. After each query is read, the program should answer the query, namely the y -coordinate of the segment which is vertically above the query point. All answers should be written into a file named “answers.out”.

1.1 “input_segments.inp” format

The first line is the number n of segments in the file. Each of following n lines consists of three numbers x_1, x_2, y , where $x_1 < x_2$, all numbers are integers. Each triple of numbers describe a single input segment, whose endpoints has coordinates (x_1, y) (the left endpoint) and (x_2, y) (the right endpoint). For example

3
30 100 900
64 200 799
140 292 500

1.2 “queries.inp” format

Here every line is a pair of integers, describing the x and y coordination of a query point. For example

70 49
241 2083
1000 20
130 28

Note that the number of lines is not specified.

1.3 “answers.out” format

This file contains a line for each line in the “queries.inp”. An answer is specified by the y -coordinate of the segment which is above the query point. For example.

The segment vertically above the query point (70,49) has y-value 799
There is no segment about the query point (241,2083)
There is no segment about the query point (1000,20)
The segment vertically above the query point (130,28) has y-value 799