Practice 11 – Linked List Implementation & its Applications

(12/13)

1. **Scan line**

    Sort processed objects with a key and then use the key as the metric of scan line to sweep all objects under processing in sorting order. During sweeping procedures, you can do any operation to realize the function you want. This kind of incremental operations can save much run time.

    A. **Example. Line overlap problem: sort all lines and scan all lines. During sweeping sorted lines, compare line overlap conditions for local lines.**

2. **Problem 1 (75 pts)**

    Relation checking is to examine if any two objects are related in terms of some relational metric. One visibility checking in geometry problems is to examine if one geometrical object can "see" another object. For instance given a set of vertical lines, a line is said to be able to see another line if the projections of two lines (*l*1 and *l*2) onto the *y*-axis overlap and the accumulated projections onto the *y*-axis of the lines located between *l*1 and *l*2 do not contain the overlapping projection of *l*1 and *l*2 completely. Figure 1 shows the concept of projection lines. **The direction of projection light is from left to right, i.e., we only check if the lines right to a line can be seen by the line. Notably if the projects of two lines overlap at only a point, they are not regarded as visible.** Implement a program to report the visible lines for each line.
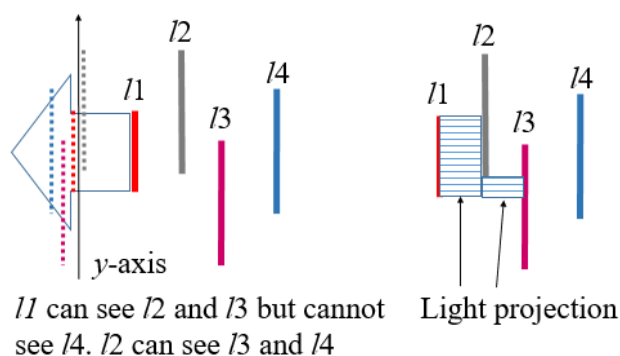

    l1 can see *l2* and *l3* but cannot see *l4*. *l2* can see *l3* and *l4*   Light projection

    Fig. 1

Input Format:

    There are four integers in a line, where the first integer is the line id, the second integer is its *x*-coordinate and the third and the last ones are the bottom and top *y*-coordinates. Two integers are separated by space character. Notably for all lines at the same *x*-coordinate, they may overlap.

Input example 1: there are five vertical lines and their input content is as follows.

```
1   100   75    325
2   5   5    120
3   200   100   255
4   75   325   500
5   300   20    50
```

Output format: Please print out the visible lines for each line. Print the report to a file named "vLine.out"

Output example: The output file "vLine.out" for the example in Input example 1 looks like:

```
1    3
2    1    5
3
4
5
```

Each line contains the visible lines for a line. The first integer is the line id and then the following is the visible line ids to current line. Two adjacent integers are separated by a space character. If a line has no visible line, then just print its line id at its row.

Main features:

A.   Read all vertical lines from lineSet1.txt and sort all lines in increasing order of *x*-coordinates and then *y*-coordinate if a tie occurs at *x*-ccordinate using qsort() (due tonight).

B.   Report vLine.out (due in the midnight of 12/20)

C.   Complete Problem 1 first, and then have the right to work on Problem 2.