CSE 241 Programming Assignment 7 (Grade weight is twice as much as of a regular assignment)

DUE

June 10, 2018, 23:55

Description

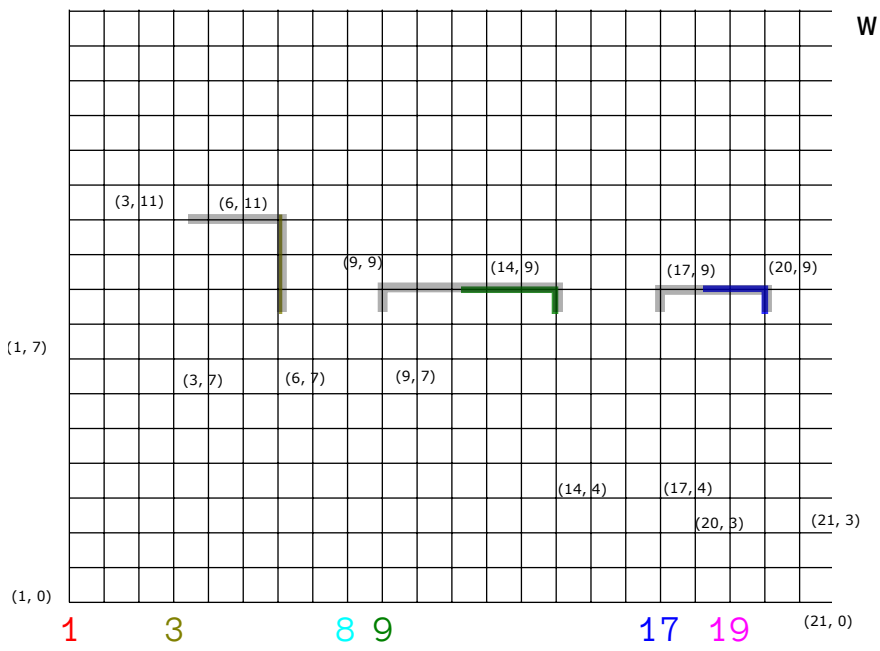
- This is an individual assignment. Please do not collaborate
- If you think that this document does not clearly describes the assignment, ask questions before its too late.

This is a `Java` Programming assignment. You will write a `Java` program which solves a problem known as `The Skyline Problem`.

- Your program reads a file:
  - `data.txt`
- According to content in `data.txt`, the program utilizes necessary classes and solves the problem described below.
- Your program prints the solution to the standard output and at the same time creates a simple graphics window for the result.

The Skyline Problem

From a distance, the view of a collection of high-rise buildings reveals a profile. With sufficient simplifications, this observation turns into a problem called the skyline problem. The objective is to find the profile created by the roofs of the buildings. The simplifications are: All of the buildings are represented by rectangles and the profile is on 2D plane.



width	height	position
3	11	3
10	7	1
5	9	9
12	4	8
3	9	17
2	3	19

Figure 1: Visualization

`data.txt`

- This file holds information about rectangles. Each line is related with a rectangle.
- Fields are all integers
- Fields are separated with spaces
- Format:

```
<width> <height> <position>
<width> <height> <position>
<width> <height> <position>
   .
   .
   .
```

- Example:

```
3 9 17
5 9 9
12 4 8
3 11 3
10 7 1
2 3 19
```

Output

- List all the corners of the skyline in the standard output.
- Make sure it is ordered. (i.e. when followed, creates a path of the skyline)
- Example:

```
1 (1, 0), (1, 7), (3, 7), (3, 11), (6, 11), (6, 7), (9, 7), (9, 9), (14, 9), (14, 4), (17, 4),
  ↪ (17, 9), (20, 9), (20, 3), (21, 3), (21, 0)
```

Graphics Window

- Show the `skyline` in a simple graphics window.
- Create a `JPanel`, `JFrame`, other necessary components and draw lines between the points listed in the solution.
- Pay attention to the coordinate system origin.
- If necessary, scale the drawing. (A path spanning a couple of pixels cannot be visible)

Remarks

- There is no limit on the number of rectangles.
- Rectangles are **not** sorted.
- The same rectangle can appear more than once in `data.txt`.
- Assume there isn't any errors in `data.txt`.
- Grade weight is twice as much as of a regular assignment. In fact, this is two assignments combined in one.
- `The Skyline Problem` is a well known problem and you can find a lot of help if you search the internet. But, this does not mean that you can copy an existing code. Try to understand the algorithm and come up with your implementation.
- A script will compare your code not just against your classmates' submissions but against other code collected from different web pages.
- Provide a file(`comments.pdf`) which includes your comments about the program you implemented. Discuss its efficiency and make comments about its time complexity.

Turn in:

- Source code of a complete Java program and a suitable makefile. You don't need an IDE for this assignment. If you use an IDE, do **not** include any files related with it. I should be able to compile and run your code in a command window.
- `comments.pdf` (Described in Section: `Remarks`)
- A script will be used in order to check the correctness of your results. So, be careful not to violate the expected output format.
- Provide comments unless you are not interested in partial credit. (If I cannot easily understand your design, you may loose points.)

- You cannot get full credit if your implementation contradicts with the statements in this document.

Late Submission

- `(0,24]` hours: `-20%`
- `(24,48]` hours: `-40%`
- `(48,72]` hours: `-60%`
- `(72,-)` hours: `-100%`