COMP 610 Project 2: Stable Matching Equity

Due: Friday December 14 at 2355

Points: 30 points possible

Overview: Your program will be given an instance of StackingBoxes. You will design and code a Dynamic Programming solution. As a reminder, the input to StackingBoxes is the length, width, and height of a set of boxes. Your goal is to stack some of the boxes up without reorientation (ie don't turn or flip the boxes) so that the stack is as tall as possible. Recall that for box i to sit on top of box j, $l_i < l_j$ and $w_i < w_j$.

Details: The input will come from a file called input.txt which will be placed in the same directory as your java file. The first line of the file will have a single integer value N which will be the number of boxes. The next N lines will be the whitespace separated dimensions of each box. See the sample input below for examples.

Design and run a Dynamic Programming algorithm to find the height of the tallest stack of boxes you can make with the input.

You can discuss ideas for the algorithm to be used with anyone and consult any source (books, internet, etc). However, for this project, you are expected to write the code on your own with limited or no assistance from the professor, no assistance from others, and limited or no assistance from other sources (books, internet, etc). To clarify, you can seek assistance in understanding the task and how it can be solved, but "your code" should be written by you: not written by others, not copied from others, not copied from books/internet.

Picky, but required specifications: Your project must:

- be submitted via canvas.
- consist of 1 or more dot-java files (no class files, zip files, input files or other files should be submitted). Each file must have your name and which project you are submitting as comments on the first 2 lines.
- not be placed into any package (for the java pedants, it must be in the default package).
- have one file called Project2.java.
- compile using the command 'javac Project2.java'.
- run using the command 'java Project2',
- accept input from a file called input.txt in the same directory as the java file(s) formatted precisely as described above.
- be submitted on time (early and multiple times is fine).

If your project fails any of the above, you will receive a zero (recall that each of you may replace 1 and only 1 project that receives a zero this semester). If your project meets the requirements above then it will be graded on whether it:

- is designed and formatted reasonably (correct indentation, no excessively long lines, no excessively long methods, has useful method/variable names, etc) and
- accomplishes the goal of the project. In other words, the output should be the correct answer, computed in a valid way, formatted correctly.

Sample execution: If input.txt contains

```
5
1 2 3
3 4 2
2 5 1
4 5 5
6 3 2
```

then the output should be

```
10
```

because you could stack up the boxes 1, 2, and 4.

If input.txt contains

```
5
5 5 4
1 7 2
2 6 3
3 5 4
4 4 5
```

then the output should be

```
9
```

becaus you could stack up boxes 1 and 5 (with 1 underneath).

Stray Thoughts:
I suggest you finish and submit your project at least several days in advance. This way you have time and opportunity to ask any last questions and verify that what you upload satisfies the requirements. There is nothing wrong with working on the project for a day and uploading your code, working for another day and uploading your improved code, ..., working for another day and uploading your final version. In fact there are advantages: you have a fairly reliable place that keeps your versions and even if you get busy at the last moment you have still uploaded your best version.
Your project should be written and understood by you. Helping or receiving help from others to figure out what is allowed/required is fine, but copying code is not. Significant shared source code indicates that you either did not write/understand what you submitted or you provided code to another (allowing them to submit code they did not write/understand). Both are academic dishonesty.