ASSIGNMENT 4 EX-TRA CREDIT

Data Structures

CSI310

Spring 2011

DUE: On Blackboard, Friday, Feb 11 at 11:59PM. Upload your .java file(s) as attachment(s).

Create a complete Java application with the following structure:

1. It must first import four (4) images from the user. It may store references to them in an array, or in 4 simple references, or any other way you see fit.

2. It must display (using the show() method of the Picture class) one image composed of the four imported images so that the first two are above and the last two are below.
   The first must be left of the second and the third must be left of the fourth.
   The imported images must NOT be resized, overlapped or clipped in any way.

3. Some of the images can and MUST be touching, that is, no space in between.

4. It MUST work with any combination of reasonably sized images and with any combination of the same or different aspect ratios. The aspect ratio of an image is the ratio of its width to its height. Thus tall, skinny images have low aspect ratios, squares have aspect ratio one, and short, long images have high aspect ratios. To “work” of course means not crash and display the result image that meets the specification.

5. Now here is the hard part: A successful effort must be made (for getting the extra credit) to lay out the images in an efficient and pleasing manner. An algorithm that gives BETTER results than the results shown below must be invented and implemented.

   Unlike in many other school CS assignments, this requirement is not well defined and many different solutions are acceptable. In fact, an analysis or some literature research attempting to find solution criteria or best methods has (purposely) NOT been undertaken by the professor (at least yet). This assignment is like those you will get professionally.

Below is a sample of the results from the most simplistic and obvious method that I can find that works on all combinations of image sizes and aspect ratios. Your solution must do better!

Your’s must do better on the 3rd example, might improve the above two, and also work on all other inputs.