Common bugs for rastering:

- Calling getGraphics() more than once.
    - Actually this is okay if you draw your lines on the second image. What's possible is that you're actually drawing on the first getGraphics(), so the stroke color/etc isn't set, so it'll be a very pale line.
- Forgetting to check during quadtree construction for both overlap of the query rectangle AND that the node is a leaf node.
    - Using some provided implementation of QuadTree, such as the one provided on Princeton. It doesn't lend itself nicely to what we need in this assignment.
- Calling imageio.write before building the appropriate buffered image.
    - More specifically, calling imageio.write in every iteration of the for loop in which they add each image tile into a "total" buffered image.
- Using > instead of >= when writing images out to the output stream.
- Overly complex rectangle logic leads to ugly bugs.
    - Try something like http://stackoverflow.com/questions/306316/determine-if-two-rectangles-overlap-each-other
    - BE SURE to account for how java's y-coordinate system increases in y-value as you go downward, while latitude decreases as you go downward!
- Parameter bugs:
    - Confusing raster_ul_lon with ullon. Your returned raster_ul_lon (and lat) should be the actual frame shown, and your widths and heights should be multiples of 256.. Make sure your return values match the values given in the sample html files.
    - As listed in the spec, your returned width and height values must be integers, not longs, floats, doubles, or anything else. You can use the intValue method to convert a Double into an int.
- If you're getting "output image output file size not same in bytes" from the local autograder, and you did something weird to set up your images that you learned from stack overflow: Make sure you're not including an alpha channel in your image.
- Treating longitude (lon) as the y-coordinate and latitude (lat) as the x-coordinate, when it should be the other way around.

Common bugs for paths:

- If your drawn path looks slightly wrong, note that the distance per pixel in the latitude and longitude directions may be different!
- When doing A*, if your implementation explicitly marks vertices, this should be done so AFTER THE VERTEX IS DEQUEUED, not after it is added.