1. This is one of the provided test cases.) Submit a corresponding image from your region growing algorithm. Depending on choices of parameters, it may be somewhat different from mine; that's okay. Briefly describe the implementation and parameter choices you made and their impact on the detected regions.
    a. In our implementation of the region-growing algorithm using RegionFinder, we focused primarily on tracking pixel regions based on color similarity. The necessary parameter utilized overall was the parameter targetColor, which was obtained from an outside input that necessitates a random color that the algorithm looks for, along with pointColor, one of two parameters implemented in the colorMatch method to make sure that the two colors were a given closeness in color. The algorithm explores unvisited pixels that match the target color with regard to maxColorDiff, and utilizes a breadth-first search. This connects neighboring pixels and reassigns random colors to all detected regions for user visualization. We also included the getLargestRegion method, which found the largest region from the list of regions and returned it for future reference
2. Now plug that into the provided webcam-based scaffold. Set the target color by mouse press, and paint according to the detected brush (largest region). Submit screenshots of you / your partner's work, both an image of webcam with recolored regions and a resulting painting. Briefly describe the utility and limitations of region growing in this context.
    a. For the implementation of region-growing in the context of the webcam, we allowed users to select a target color by clicking on the webcam feed via the handlePress method. The RGB value of the pixel that is clicked is then captured, and the regionFinder class identifies the largest region matching said color. That region is recovered using the largestRegion method, and recolored on the camera feed for visibility with a given brush color.

       The utility of region growing in this context is the ability to detect areas of interest, which gives users the ability to edit images via their own input. However, there are indeed limitations. The algorithm is contingent only on the chosen parameters maxColorDiff and minRegion, as their numbers can lead to issues concerning what the actual largest Region is, what region should be considered, and what colors should be considered. Proper calibration of these instance variables is crucial to navigating the code correctly. Also, there were noticeable issues with the getting of the pixels depending on light exposure, especially if backgrounds were highly variant with lighting in general. Optimizing these limitations will be very necessary to further improve the program.