In this machine problem we had to implement the CCL algorithm to identify and count the number of distinct regions in a binary image. According to the assignment this was done in MATLAB, which was a bit difficult as MATLAB didn't quite have working set tools (inputs must be a character vector error, I'm looking at you!). So, to get around this problem, three functions were designed, the CCL function as requested, a function to determine if an element is a part of a set represented as a MATLAB array, and a function that unions two sets within a provided superset that was represented as a MATLAB cell array and their indices within this cell array.

The CCL algorithm can further be broken down into parts, the first part being initialization of certain variables, the second part being iteration over the image searching for regions and creating the equivalence table, and the third part using the equivalence table to reassign regions and recolor the final image using the index of the set within the equivalence table's cell array to match to the linear space between 0 and 255 created by knowing the number of regions (the number of arrays {sets} in the cell array {the superset/equivalence table}).

The final result produced the greyscale images turned in along with this report, as well as the number of regions in each image. In test.bmp, there was only a single region. In gun.bmp, there were 4 distinct regions. In face.bmp, there were 6 distinct regions, and in face_old.bmp there were 10 distinct regions (I didn't believe this at first and almost had a heart attack, until I noticed the slight disconnects within what had looked like one solid region thanks to the greyscale image produced by CCL).