Sam Kelso

Explanation of “fill()” Function

This implementation of the fill function uses a recursive algorithm. Taking the image and seed_point as inputs, the function first checks whether the arguments are valid. This involves checking the image is non-empty, the seed_point coordinates are integer values and that the seed_point lies within the bounds of the image. We also check whether the seed_point lies on an edge within the image (i.e that at this index of the list object there is a 1 or not). In each case if the input of the function does not pass the check then fill() returns the input image.

If all these criteria are satisfied the algorithm then checks that the seed_point has not already been filled, thus equal to zero. This is to avoid an infinite loop where the algorithm continues to fill points of the image that have already been filled. If the seed_point is equal to 0 then the function fills this point by setting the relevant point in the image list to 2.

The algorithm then continues recursively applying the fill() function to the points one above, left, right and below the seed_point updating the image being filled. If any of these points are not valid points to be filled, then the function returns the same image therefore not continuing in that direction. This process repeats recursively resulting in the final image returned being filled as required.