# Lab 4: Hilbert curve and images

Due date: Sunday Feb 25, 11:59pm

## 1   Hilbert curve [1oo points]

Study this website: http://www.fundza.com/algorithmic/space_filling/hilbert/basics/index.html (I put a PDF of the site in Canvas). They discuss the Hilbert curve and how to generate it, we will write a Python code to do that (actually there is a python code in the websited, but it is written for a particular rendering system (which we do not use), we use simply matplotlib. The key will be that where the pseudo-code suggest to do a **LineTo** we will append the coordinates to a global list. This list will then be plotted. The hilbert curve fills the space and with a recursion level of 7 we reach a nice plot where we still can see what is going on (Trying to do curves with levels ¿8 will lead to long wait times and black ouput because there are too many lines on the graph), Here a few examples (with recursion level 5, and 7):
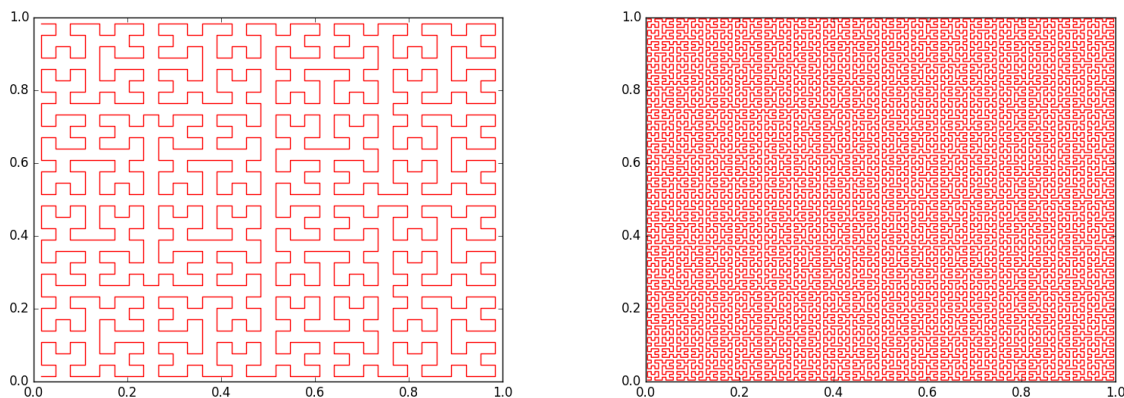
Write a python file that can take the recursion level as commandline input, the code should allow values between 1 and 8 and reject others. The it will use a modified hilbert function that can generate a list of x,y coordinate pairs. These pairs have to be converted to an x and y vector and the y,x vectors will then be plotted using matplotlib, and finally save the result as pdf and call it hilbert#.pdf where the # is the recursion level used to generate the picture.

## 2   Hilbert curve coloring [50 points]

This section is optional (if you have solved the first problem within a short time or want to be challenged generate the figure below (it uses a frog picture and the hilbert function from section 1. The image is read into matplotlib (using imread), the hilbert curve is then chopped up into pairs of coordinates along the line e.g. $[[x_n, x_{n+1}], [y_n, y_{n+1}]]$ and each segment is plotted with the color vector at location $x_n, y_n$, the picture uses RGB as the color vector and every pixel is small list with three values $[R, G, B]$, there are several additional challenges (direction of Y axis, Hilbert curve fills the rectangle with coordinates [[0,0], [1,0],[1,1],[0,1]] but the pictures has a different size [hint: map the hilbert coordinate space to the image coordinates], to make the background black (I used the function plt.fillbetween(), and adjusting the linewidth).