Problem 1. (Space Filling Curves)

a) Compute the Z-values of regions A and B in the figure below, assuming that \( K = 4 \) bits \((2^4 = 16\) values per axis\) and \( A1=(5.5, 6.5), A2=(8.5, 5), A3=(9.1, 9.2), A4=(7, 10), B1=(0.4, 1.8), B2=(1.2, 0.6), B3=(4, 0.8), B4=(4.5, 2.3) \) and \( B5=(0.9, 3.2) \).

\[Z1 = \ldots, \quad Z2 = \ldots, \quad Z3 = \ldots, \quad Z4 = \ldots, \quad Z5 = \ldots, \quad Z6 = \ldots, \quad Z7 = \ldots, \quad Z8 = \ldots, \quad Z9 = \ldots, \quad Z10 = \ldots\]

b) Write a program that computes the z-value and the Hilbert-value of a 2-d pixel, as well as the inverse (given a z-value or a Hilbert value returns the pixel.) The program should take as input the precision per dimension (or the order of the curve) and the \( x \) and \( y \) values, or for the inverse, the z- or Hilbert-value of the pixel. A small test file will be provided to you for testing the correctness of your program. Note that you should hand in the source code of your program via email.

HINT: Methods to compute z- and Hilbert- values can be found in the following paper:
H. V. Jagadish: Linear Clustering of Objects with Multiple Attributes. ACM SIGMOD Conference 1990, pages 332-342.

Problem 2. (R-trees)

Let \( D \) be a 2-dimensional point dataset and \( p = (x, y) \) a point in that set. The coordinates of all points are positive. Consider the function: \( f(p) : D \rightarrow R \), where \( f(p) = a_1x + a_2y \) and \( a_1 + a_2 = 1 \). The values for \( a_1 \) and \( a_2 \) are given by the user. The idea is that each user gives different importance (weight) to different attributes. We want to find the point (or points) that maximize this function. This type of queries are called preference queries. Now, assume that an R-tree is used to store the dataset \( D \).

(a) Design an efficient search procedure that uses the R-tree to find the point(s) that maximize the function \( f \). Give the pseudo-code of the algorithm and explain how it works.

(b) What is the property that allows the design and guarantees the correctness of your algorithm?
Problem 3. (General)

Answer the following questions about some of the methods that we discussed so far:

1. What are the advantages and disadvantages of the R*-tree compared to the R-tree. You can discuss performance, specific query types, other overheads, etc.

2. Discuss the different merging policies of the Grid File and what are the pros and cons of each method. Which method you would use in your implementation? Why? Explain.

3. Propose an algorithm to compute efficiently spatial join queries between two spatial relations that contain regions, assuming that both relations have been indexed by two Linear Quadtrees. State all the assumptions that you make.