Exercise 1: Hierarchization in Higher Dimensions

In this exercise we will implement the multi-recursive algorithm for hierarchization of a multi-dimensional regular sparse grid. The structure of the code resembles strongly the one-dimensional case, and so we again have a class \( \text{PagodaFunction} \) representing our grid points.

(i) Implement the refinement criterion \( \text{MinLevelCriterion} \) that adds all points up to a specified level to a given grid.
   
   **Hint:** In your grid traversal, try to avoid multiple visits to the same grid points.

(ii) Implement the function \( \text{hierarchize} \) efficiently using a recursive approach.
   
   **Hint:** The underlying traversal algorithm can be implemented similar to the one in (i).

(iii) Implement a function to compute the volume of the sparse grid interpolant.

Exercise 2: The Combination Technique – A Different View on Sparse Grids

Dealing with hierarchical bases often turns out to be sophisticated. On this worksheet we will therefore see how the so-called combination technique finds a sparse grid interpolant, that approximates a function on a number of full grids, each consisting only of a “relatively small” number of grid points.

Let \( u_l \) for \( l \in \mathbb{N}^2 \) be a function \( u : [0, 1]^2 \to \mathbb{R} \) interpolating piecewise bilinearly at the inner grid points, at the boundary \( u \) is assumed to be zero again.

(i) \( V_l \) can be decomposed into a set of subspaces \( W_l \). Accordingly, the interpolant \( u_l \in V_l \) can be written as a sum of \( w_l \in W_l \).

   Spot the grid associated with \( u_{3,2} \) in the right part of Figure 1. Identify those subspaces in the left part that are needed to reconstruct \( u_{3,2} \).
Figure 1: The two parts in the picture show the grid points and supports associated with interpolants \( w_l \) (left) and \( u_l \) (right) up to level 4 for the 2d case.

(ii) Use the result from (i) to rewrite

\[
\sum_{|l|_1 = n+1} u_l, \quad n \in \mathbb{N}
\]

for the two-dimensional case as a weighted sum of \( w_l \).

**Hint:** Look at the subspace scheme in Figure 1 and count the occurrences of each subspace in the sum. What do you notice when comparing \( w_l \) with common level \( n = |l|_1 + \text{dim} - 1 \)?

(iii) In the final step use the previous results to give a representation of the sparse grid interpolant

\[
u_n^D := \sum_{|l|_1 \leq n+1} w_l
\]

as a weighted sum of \( u_l \). Again, count the occurrences of the \( w_l \).

(iv) Assume you are talking to a person who knows how to approximate the volume \( F_2(u) \) through the trapezoidal rule (in 2d) with respect to \( u_l \). Give instructions on how to write a program that implements a sparse grid approximation of \( F_2(u) \). Remember Archimedes quadrature.

(v) Compare this method with Archimedes quadrature — what are the (dis-)advantages?