[CS 11 25.1] Lab 1e – Hilbert's Curve

Submit solution

✓ **Points:** 130 (partial)
Time limit: 4.0s
Memory limit: 1G

Problem type

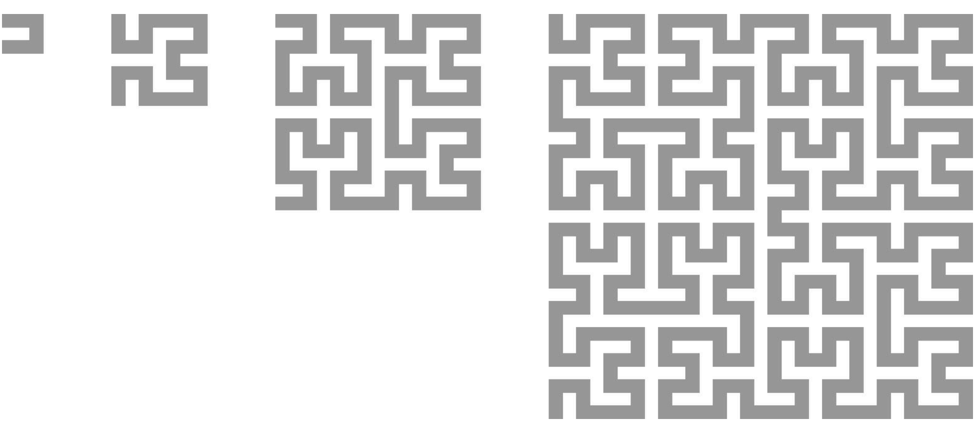
Allowed languages
~~NONE~~, py3

Cheatsheet is available here: https://oj.dcs.upd.edu.ph/cs11cheatsheet/

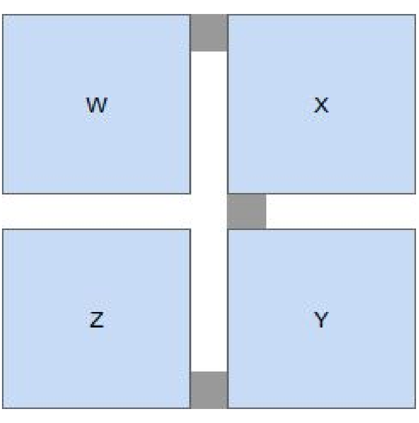
Problem Statement

The mathematician David Hilbert defined an interesting infinite sequence of curves $C_1, C_2, C_3, \ldots$.

The following illustrates C_1 thru C_4.



More precisely, for each $k \geq 1$, curve C_{k+1} is obtained from four copies of curve C_k using the following diagram:



where:

- X and Y are copies of C_k,
- W is a copy of C_k rotated 90 degrees clockwise,
- Z is a copy of C_k rotated 90 degrees counterclockwise, and
- the gray squares serve as "connections" between the pieces.

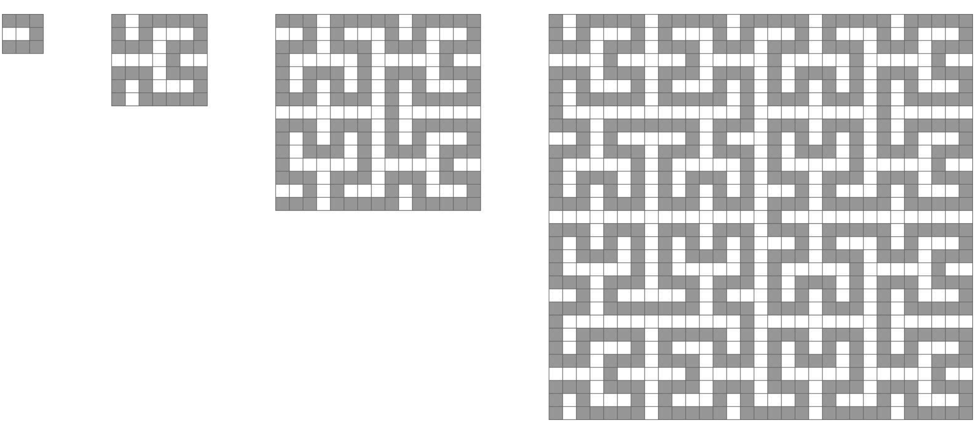
The curve C_1 itself is the "base case" and is just directly defined as the leftmost curve in the first image above.

Notice that each of Hilbert's curves is a single continuous path from the top-left cell to the bottom-left cell of a square grid!

Hilbert showed that if the C_ks are scaled down to fit somewhat snugly into a unit square, then the *limit* of the curves is a curve that *fills* the whole unit square—in other words, a "one-dimensional" object completely filling a "two-dimensional" object!

Your task is to draw a subgrid of the curve C_k.

Here are C_1 thru C_4 again but with the gridlines:



Task Details

Your task is to implement a function called `hilbert_subgrid`. This function has five parameters, all `int`s: k, i_1, j_1, i_2, j_2.

The function must return a `tuple` of $i_2 - i_1$ `str`s. Each of these `str`s must be of length $j_2 - j_1$. This `tuple` of `str`s represents the subgrid with topleft corner at row i_1 and column j_1, and each character is `#` if the Hilbert curve passes through that cell, and a space (` `) otherwise.

We number the rows from top to bottom consecutively starting from 0, and we number the columns from left to right consecutively starting from 0.

Restrictions

For this problem:

- Assignment is allowed.
- Recursion is allowed.
- Up to 6 function definitions are allowed.
- Comprehensions are **disallowed**.
- `range` is **disallowed**.
- The `abs` symbol is now allowed.
- The source code limit is 2000.

Example Calls

Example 1 Function Call

```
hilbert_subgrid(1, 0, 0, 3, 3)
```

Copy

Example 1 Return Value

```
(
    '###',
    '  #',
    '###',
)
```

Copy

Example 2 Function Call

```
hilbert_subgrid(3, 1, 2, 6, 8)
```

Copy

Example 2 Return Value

```
(
    '# #   ',
    '# ### ',
    '    # ',
    '### # ',
    '# # # ',
)
```

Copy

Constraints

- The function `hilbert_subgrid` will be called at most 50 times.
- $1 \leq k \leq 100$
- $i_2 - i_1 \leq 30$
- $j_2 - j_1 \leq 30$
- $0 \leq i_1 < i_2 \leq r_k$ where r_k is the number of rows of C_k.
- $0 \leq j_1 < j_2 \leq c_k$ where c_k is the number of columns of C_k.

Scoring

Note: New tests may be added and all submissions may be rejudged at a later time. (All future tests will satisfy the constraints.)

- You get 130 ● points if you solve all test cases.

Clarifications

Report an issue

No clarifications have been made at this time.