[CS 11] Prac 8p – Maze Drawing IV

Problem Statement

Given a maze, please draw its *skeleton*. For example, if the input maze is:

```
##################
#   #   #        #
### ### ### ######
#   # #          #
# ### # # ########
# # # # #        #
# # # ### ########
# # # # # #      #
# # # # # # ######
# #     # #      #
# ### # # # ### #
#     #        # #
##################
```

then the required output is:

```
+-+ +-+ +-+---+
  |   |   |
+-+ + +-+-+---+
|   | | |
| + | + +-----+
| | |   |
| | | + | +---+
| | | | | |
| +-+-+ | +---+
|   | | | |   |
+---+ +-+-+-+ +
```

The skeleton denotes to the *paths* inside the original maze!

Submit solution [CS 11] Practice 8

Points: 150 (partial)
Time limit: 4.0s
Memory limit: 1G

Author:
kvatienza (Kevin Atienza)

Problem type

Allowed languages
~~NONE~~, py3

Task Details

Your task is to implement a function called `get_maze_skeleton`. This function has a single parameter: a `tuple` of `str`s.

The function must return a `list` of `str`s representing the maze skeleton.

Restrictions

(See 8a for more restrictions)

For this problem:

- Loops and lists are allowed.
- Up to 18 function definitions are allowed.
- Recursion is **disallowed**. (The recursion limit has been greatly reduced.)
- Sets and dictionaries are allowed.
- Generators and comprehensions are allowed.
- The source code limit is 4000.

Example Calls

Example 1 Function Call

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

Example 1 Return Value

```
[
    '+-+ +-+ +-+---+',
    '  |   |   |    ',
    '+-+ + +-+-+---+',
    '|   | | |      ',
    '| + | + +-----+',
    '| | |   |      ',
    '| | | + | +---+',
    '| | | | | |    ',
    '| +-+-+ | +---+',
    '|   | | | |   |',
    '+---+ +-+-+-+ +',
]
```

Example 1 Explanation

Hint: You can print a grid of `str`s by doing:

```
for row in grid:
    print(row)
```

or by doing

```
print(*grid, sep='\n')
```

Constraints

- The function `get_maze_skeleton` will be called at most 20 times.
- $3 \le r, c \le 75$
- The input is a valid maze.

Scoring

- You get 150 ❤️ points if you solve all test cases.

Clarifications

Report an issue

No clarifications have been made at this time.