JOHANNES KEPLER
UNIVERSITÄT LINZ

Algorithms and Data Structures 1
Summer term 2021
Dari Trendafilov
Stefan Grünberger

Assignment 3

Deadline: **Thu 06.05.2021, 23:59**
Submission via: **Moodle**

Elaboration time

*Remember the time you need for the elaboration of this assignment and document it in the file **time.txt** according to the structure illustrated in the right box. Please do not pack this file into an archive but upload it as a **separate file**.*

```
#Student ID
K12345678
#Assignment number
03
#Time in minutes
190
```

Recursion

For this assignment please submit the source code of your **`My_Maze.py`**.

1. Maze — 24 points

Implement the **`Maze`** class using the skeleton below, which aims to **recursively** find and mark the exits of a given square or rectangular maze of a size at least 3x3. Maze is represented by a string, with rows delimited by \\n:

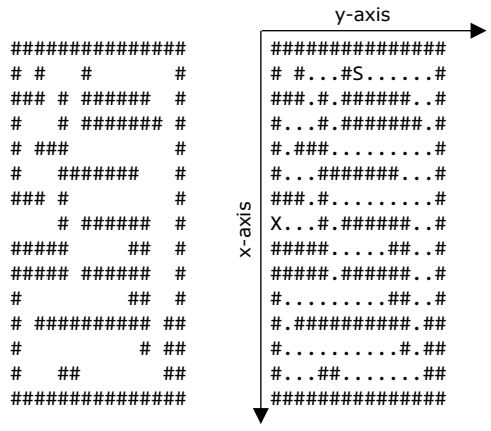
- '#' … represent obstacles/walls, which cannot be traversed
- 'S' … denote the starting position
- 'X' … mark a found exit
- ' ' … empty/blank fields represent possible paths
- '.' … dots mark visited positions/cells

```
class Maze:
    def __init__(self, maze_str: str):
    """Initialize Maze with a string, where rows are separated by newlines (\\n).

    def find_exits(self, start_x: int, start_y: int, depth: int = 0) -> None:
        """Find and save all exits into `self._exits` using recursion, save the maximum recursion depth and mark
        the maze. An exit is a path/empty cell on the outer rims of the maze.
    Args:
        start_x (int): x-coordinate to start from. 0 represents the leftmost cell.
        start_y (int): y-coordinate to start from; 0 represents the topmost cell.
        depth (int): Depth of current iteration.

    @property
    def exits(self) -> List[Tuple[int, int]]:
        """List of tuples of (x, y)-coordinates of currently found exits."""

    @property
    def max_recursion_depth(self) -> int:
        """Return the maximum recursion depth after executing find_exits()."""
```

Implement your recursive algorithm using the provided skeleton of the class **My_Maze.py** as follows:

- Search from a given starting position 'S' and mark the visited positions/cells with a dot ('.').
- Allowed moves are **north/east/west/south and the 4 diagonals (sw,se,nw,ne).**
- Search the entire maze for exits and create a list of tuples with the coordinates (x,y) of all exits. Make available this list with the property `exits()`. Mark all found exits with 'X' ('S' can be an exit too!).
- Return the maximum recursion depth of the maze exploration with the property `max_recursion_depth()`.

***Note**: Return the coordinates in the correct order as shown in the example below!*



The example shows the input (left) and the output (right) for one maze:

*) top left corner of the maze with character '#' at coordinates [0,0] is the origin
*) start position marked with 'S' at coordinates [1,7]
*) the found exit marked with 'X' at coordinates [7,0]