JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

EN.540.635 Software Carpentry

Lab 6
Maze Generation and Solver pt. 1

Begin the Lab!

- **Lab 6 (pt. 1):** Maze Generation and Solver
- **Objective:**
  - Generate (**pt. 1**) and solve (**pt. 2**) mazes using Python

So…

Lab Requirements

Files:

- Lab_6.py

Software:

- Python 3.7

Procedure

- **Mazes** are generated as png images using the Python Imaging Library (PIL), and similarly solved by reading in a png image, and outputting a new one with the solution marked out in green.
- The approach taken in this lab is to use a **Depth First Search** approach applied to maze generation and solving.
- This method starts with a stack, which is essentially a list of positions. It will be initialized at some coordinate to signify the start of the maze. As such, we may begin at: [0, 0]
- We then will look for a valid position to take, and randomly select it. Two possibilities now exist:
  1. We take a random step.
  2. No valid options exist.

Once the entire space of possible choices has been explored, it should be evident that the backtracking will continue until the position stack is empty. It is at this point that the maze generation should end.

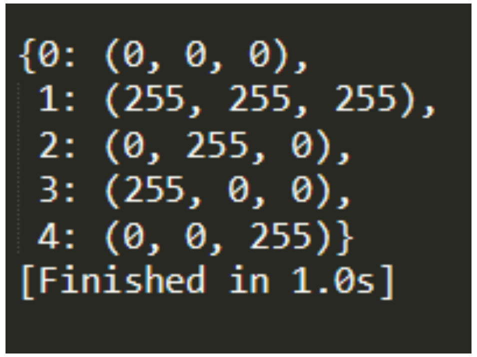
White = Path Exploration
Blue = Backtracking

Functions

- **get_colors:** Defines the color maps that the maze will use
    - Returns: A dictionary that will correlate the integer key to a color
- save_maze: Saves a maze object to a file
    - Returns: None
- load_maze: Reads a maze from a png file into a 2d list with values corresponding to the known color
    - Returns: A maze as a list
- pos_chk: Validates if the coordinates specified (x and y) are within the maze
    - Returns: True if within the maze boundaries, False if not
- generate_maze: Generates a maze using the Depth First Search method
    - Returns: None
- solve_maze: Solves a maze using the Depth First Search method
    - Returns: None

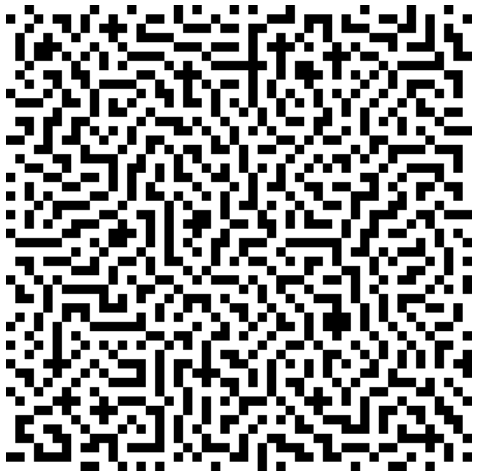
Get_colors

- 0 - Black - A wall
- 1 - White - A space to travel in the maze
- 2 - Green - A valid solution of the maze
- 3 - Red - A backtracked position during maze solving
- 4 - Blue - Start and Endpoints of the maze

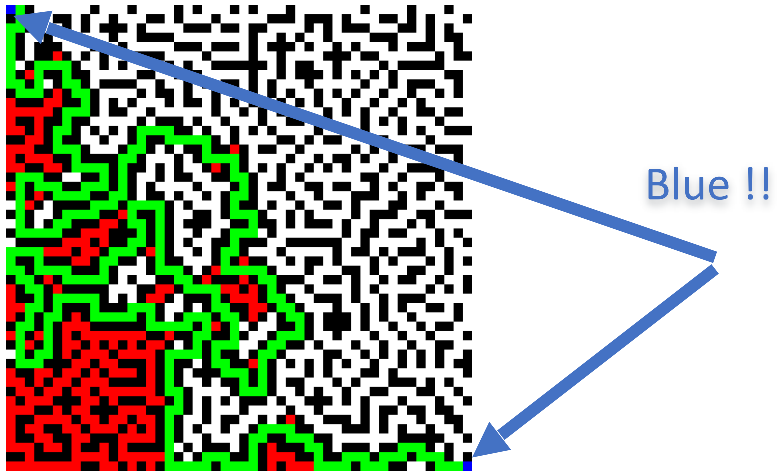


```
{0: (0, 0, 0),
 1: (255, 255, 255),
 2: (0, 255, 0),
 3: (255, 0, 0),
 4: (0, 0, 255)}
[Finished in 1.0s]
```

Get_colors ()



Generated Maze



Solved Maze

Functions

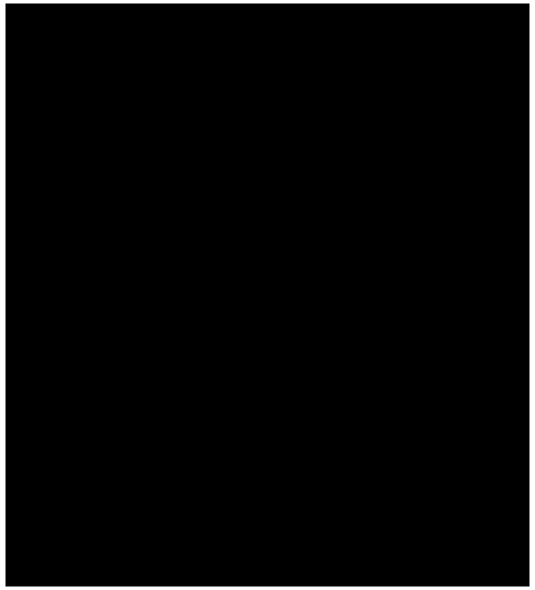
- get_colors: Defines the color maps that the maze will use
  - Returns: A dictionary that will correlate the integer key to a color
- save_maze: Saves a maze object to a file
  - Returns: None
- **load_maze:** Reads a maze from a png file into a 2d list with values corresponding to the known color
  - Returns: A maze as a list
- pos_chk: Validates if the coordinates specified (x and y) are within the maze
  - Returns: True if within the maze boundaries, False if not
- generate_maze: Generates a maze using the Depth First Search method
  - Returns: None
- solve_maze: Solves a maze using the Depth First Search method
  - Returns: None

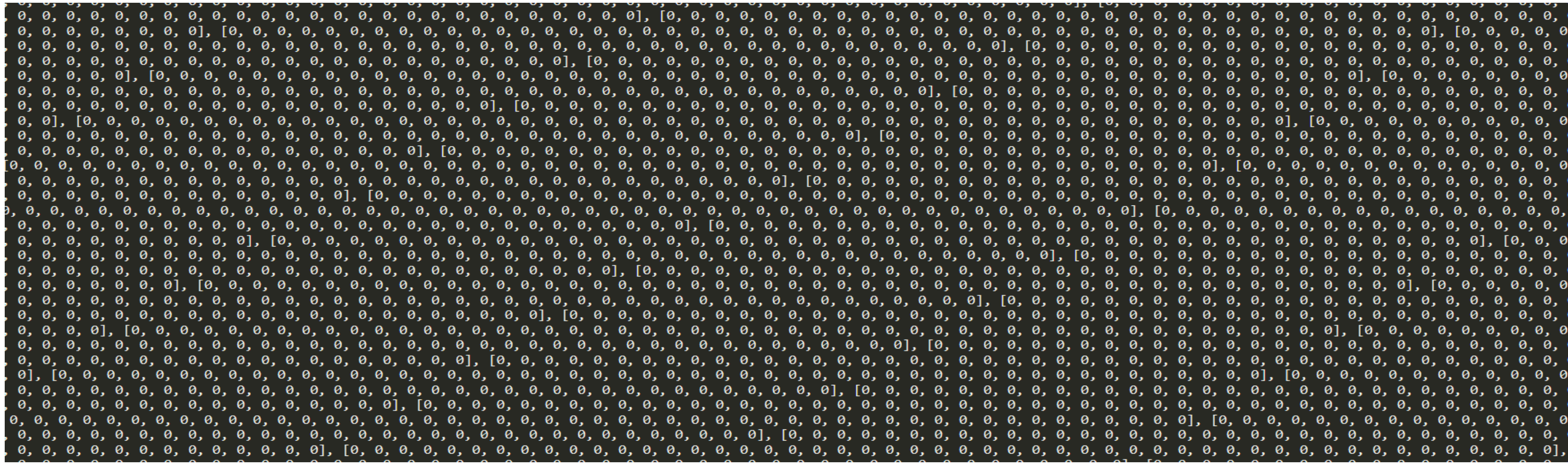
Load_maze

- **The color list is a (L X L) 2d list**
- In these examples our image size is 500 x 500
- If every block equals 10 (nblocks = 10), then every 10 x 10 pixel is 1 block
- So the length of our “stack” is 50 x 50 (L/nblocks)

Load_maze

- Here is 2d color list for an **original black** image:

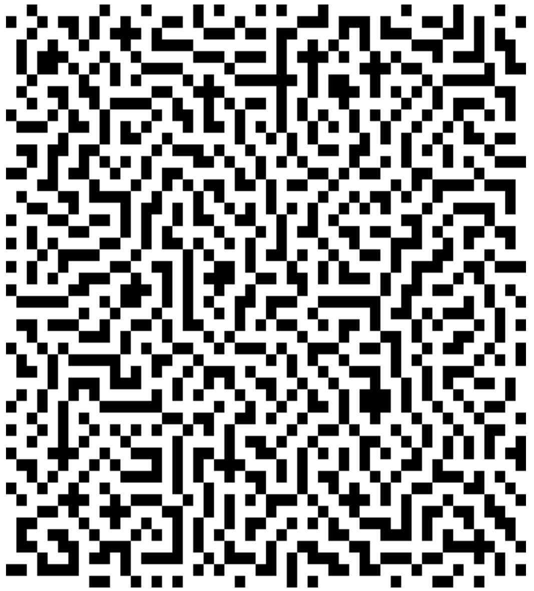


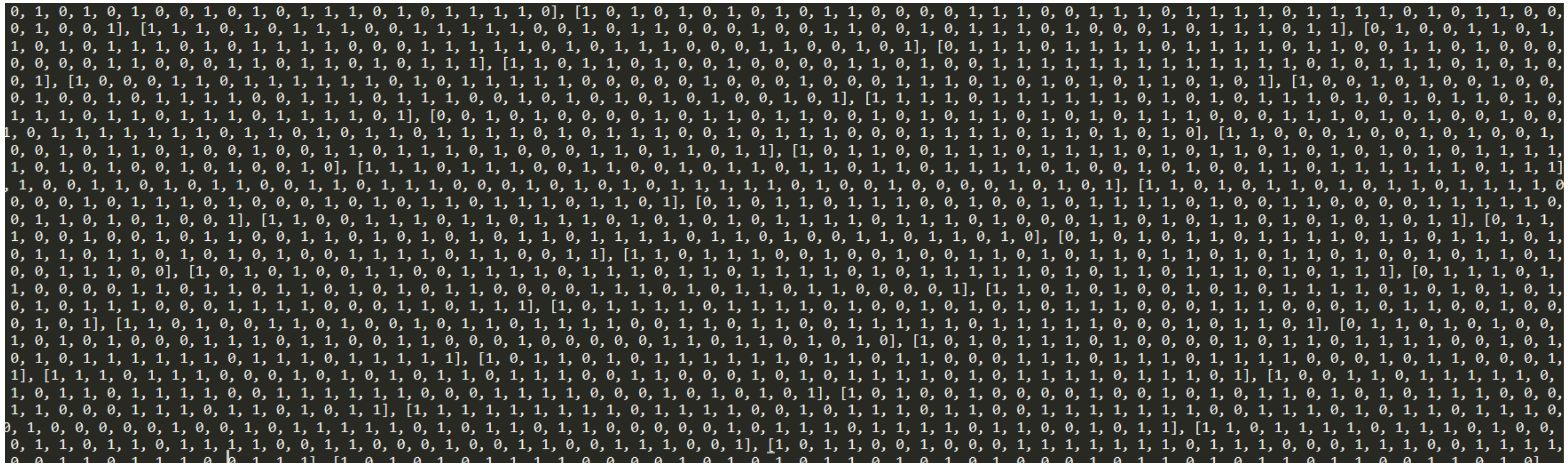


```
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0,
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0,
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0,
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0,
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0,
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
```

Load_maze

- Here is the 2d color list for a **generated** maze:

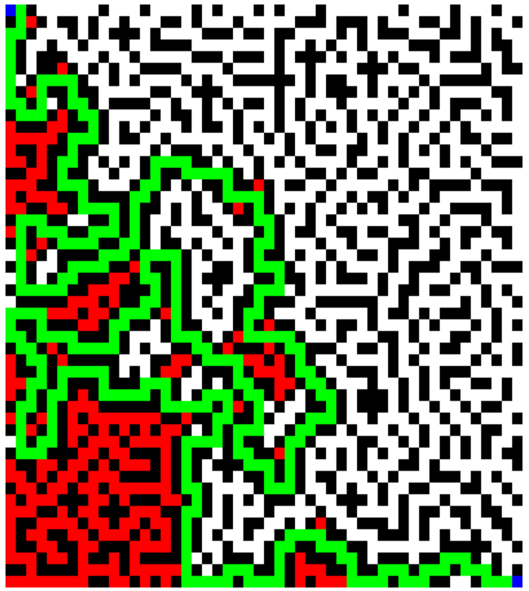


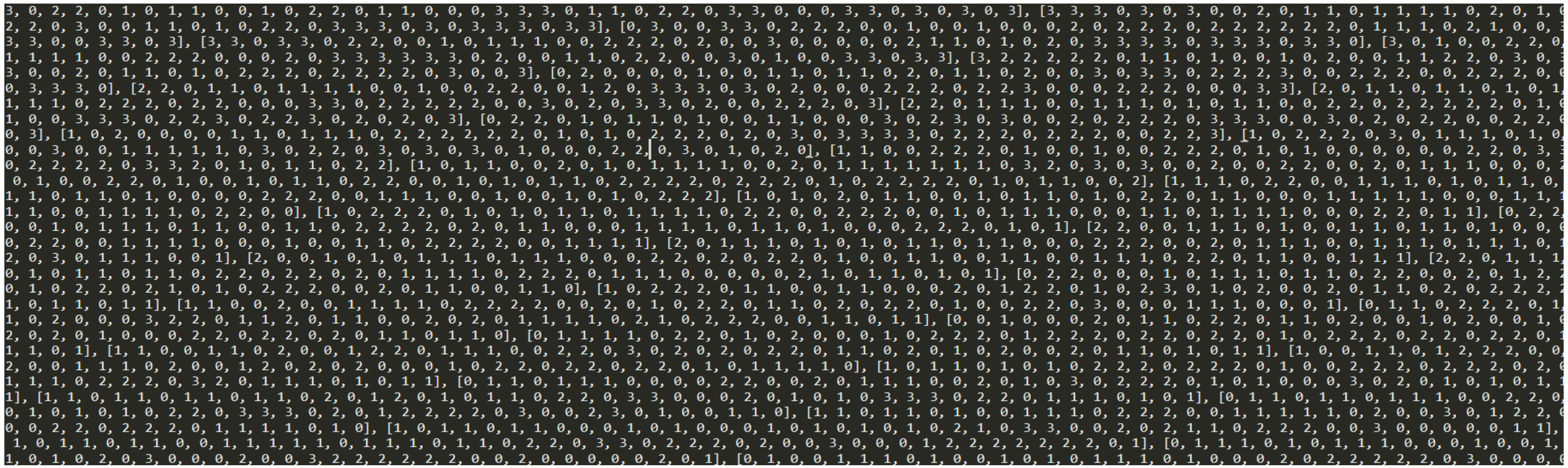


```
0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0], [1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0,
0, 1, 0, 0, 1], [1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1], [0, 1, 0, 0, 1, 1, 0, 1,
1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1], [0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0,
0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1], [1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0,
0, 1], [1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1], [1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0,
0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1], [1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0,
1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1], [0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0,
1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0], [1, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1,
0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1], [1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1,
1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0], [1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1]
, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1], [1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0
0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1], [0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0,
0, 1, 1, 0, 1, 0, 1, 0, 0, 1], [1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1], [0, 1, 1,
1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0], [0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1,
0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1], [1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1,
0, 0, 1, 1, 1, 0, 0], [1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1], [0, 1, 1, 1, 0, 1,
1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1], [1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1,
0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1], [1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0,
0, 1, 0, 1], [1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1], [0, 1, 1, 0, 1, 0, 1, 0, 0,
1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0], [1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1,
0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1], [1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1,
1], [1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1], [1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0,
1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1], [1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0,
1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0,
0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1], [1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0,
0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1], [1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1,
0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1], [1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0]
```

Load_maze

- Here is the 2d color list for a **solved** maze:





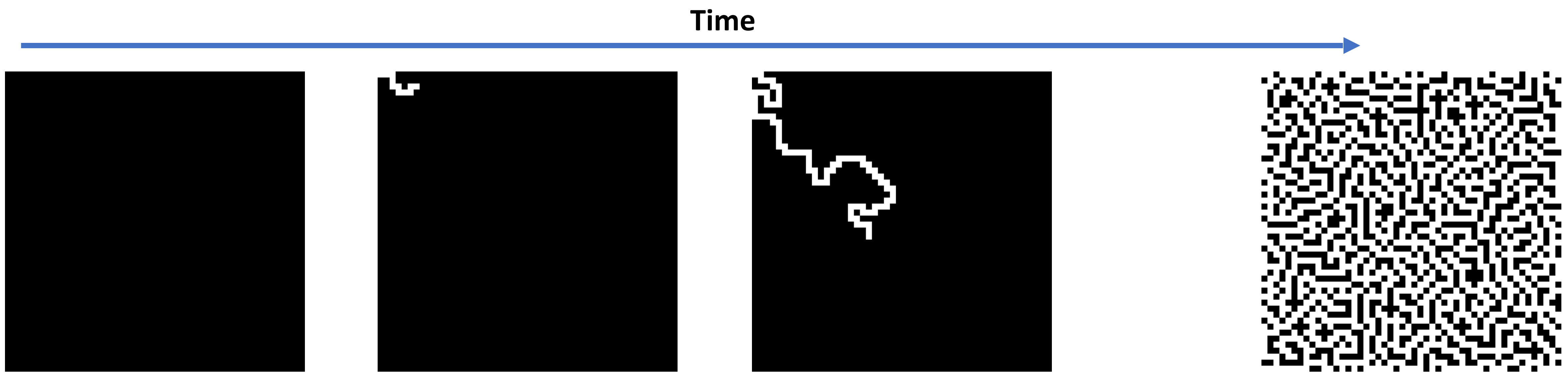
```
3, 0, 2, 2, 0, 1, 0, 1, 1, 0, 0, 1, 0, 2, 2, 0, 1, 1, 0, 0, 0, 3, 3, 3, 0, 1, 1, 0, 2, 2, 0, 3, 3, 0, 0, 0, 3, 3, 0, 3, 0, 3, 0, 3], [3, 3, 3, 0, 3, 0, 3, 0, 0, 2, 0, 1, 1, 0, 1, 1, 1, 1, 0, 2, 0, 1,
2, 2, 0, 3, 0, 0, 1, 1, 0, 1, 0, 2, 2, 0, 3, 3, 3, 0, 3, 0, 3, 3, 3, 0, 3, 3], [0, 3, 0, 0, 3, 3, 0, 2, 2, 2, 0, 0, 1, 0, 0, 1, 0, 0, 0, 2, 0, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 0, 1, 1, 1, 0, 2, 1, 0, 0,
3, 3, 0, 0, 3, 3, 0, 3], [3, 3, 0, 3, 3, 0, 2, 2, 0, 0, 1, 0, 1, 1, 1, 0, 0, 2, 2, 2, 0, 2, 0, 0, 3, 0, 0, 0, 0, 0, 2, 1, 1, 0, 1, 0, 2, 0, 3, 3, 3, 3, 0, 3, 3, 3, 0, 3, 3, 0], [3, 0, 1, 0, 0, 2, 2, 0,
1, 1, 1, 1, 0, 0, 2, 2, 2, 0, 0, 0, 2, 0, 3, 3, 3, 3, 3, 3, 0, 2, 0, 0, 1, 1, 0, 2, 2, 0, 0, 3, 0, 1, 0, 0, 3, 3, 0, 3, 3], [3, 2, 2, 2, 2, 2, 0, 1, 1, 0, 1, 0, 0, 1, 0, 2, 0, 0, 1, 1, 2, 2, 0, 3, 0,
3, 0, 0, 2, 0, 1, 1, 0, 1, 0, 2, 2, 2, 0, 2, 2, 2, 2, 0, 3, 0, 0, 3], [0, 2, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 2, 0, 1, 1, 0, 2, 0, 0, 3, 0, 3, 3, 0, 2, 2, 2, 3, 0, 0, 2, 2, 2, 0, 0, 2, 2, 2, 0,
0, 3, 3, 3, 0], [2, 2, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 2, 2, 0, 0, 1, 2, 0, 3, 3, 3, 0, 3, 0, 2, 0, 0, 0, 2, 2, 2, 0, 2, 2, 3, 0, 0, 0, 2, 2, 2, 0, 0, 0, 3, 3], [2, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1,
1, 1, 1, 0, 2, 2, 2, 0, 2, 2, 0, 0, 0, 3, 3, 0, 2, 2, 2, 2, 2, 0, 0, 3, 0, 2, 0, 3, 3, 0, 2, 0, 0, 2, 2, 2, 0, 3], [2, 2, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 2, 2, 0, 2, 2, 2, 2, 2, 0, 1,
1, 0, 0, 3, 3, 3, 0, 2, 2, 3, 0, 2, 2, 3, 0, 2, 0, 2, 0, 3], [0, 2, 2, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 3, 0, 2, 3, 0, 3, 0, 0, 2, 0, 2, 2, 2, 0, 3, 3, 3, 0, 0, 3, 0, 2, 0, 2, 2, 0, 0, 2, 2,
0, 3], [1, 0, 2, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 2, 2, 2, 2, 2, 2, 0, 1, 0, 1, 0, 2, 2, 2, 0, 2, 0, 3, 0, 3, 3, 3, 3, 0, 2, 2, 2, 0, 2, 2, 2, 0, 0, 2, 2, 3], [1, 0, 2, 2, 2, 0, 3, 0, 1, 1, 1, 0, 1, 0,
0, 0, 3, 0, 0, 1, 1, 1, 1, 1, 0, 3, 0, 2, 2, 0, 3, 0, 3, 0, 3, 0, 1, 0, 0, 0, 2, 2, 0, 3, 0, 1, 0, 2, 0], [1, 1, 0, 0, 2, 2, 2, 0, 1, 0, 0, 1, 0, 0, 2, 2, 2, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 2, 2, 0, 3,
0, 2, 2, 2, 2, 0, 3, 3, 2, 0, 1, 0, 1, 1, 0, 2, 2], [1, 0, 1, 1, 0, 0, 2, 0, 1, 0, 1, 1, 1, 1, 0, 0, 2, 0, 1, 1, 1, 1, 1, 1, 1, 0, 3, 2, 0, 3, 0, 3, 0, 0, 2, 0, 0, 2, 2, 0, 0, 2, 0, 1, 1, 1, 0, 0, 0,
 0, 1, 0, 0, 2, 2, 0, 1, 0, 0, 1, 0, 1, 1, 0, 2, 2, 0, 0, 1, 0, 1, 0, 1, 1, 0, 2, 2, 2, 2, 0, 2, 2, 2, 0, 1, 0, 2, 2, 2, 2, 0, 1, 0, 1, 1, 0, 0, 2], [1, 1, 1, 0, 2, 2, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0,
1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 2, 2, 2, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 2, 2, 2], [1, 0, 1, 0, 2, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 2, 2, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1,
1, 1, 0, 0, 1, 1, 1, 1, 0, 2, 2, 0, 0], [1, 0, 2, 2, 2, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 2, 2, 0, 0, 2, 2, 2, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 2, 2, 0, 1, 1], [0, 2, 2,
0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 2, 2, 2, 2, 0, 2, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 2, 2, 2, 0, 1, 0, 1], [2, 2, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0,
0, 2, 2, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 2, 2, 2, 2, 0, 0, 1, 1, 1, 1], [2, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 2, 2, 2, 0, 0, 2, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0,
2, 0, 3, 0, 1, 1, 1, 0, 0, 1], [2, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 2, 2, 0, 2, 0, 2, 2, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 2, 2, 0, 1, 1, 0, 0, 1, 1, 1], [2, 2, 0, 1, 1, 1,
0, 1, 0, 1, 1, 0, 1, 1, 0, 2, 2, 0, 2, 2, 0, 2, 0, 1, 1, 1, 1, 0, 2, 2, 2, 0, 1, 1, 1, 0, 0, 0, 0, 0, 2, 1, 0, 1, 1, 0, 1, 0, 1], [0, 2, 2, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 2, 2, 0, 0, 2, 0, 1, 2,
0, 1, 0, 2, 2, 0, 2, 1, 0, 1, 0, 2, 2, 2, 0, 0, 2, 0, 1, 1, 0, 0, 1, 1, 0], [1, 0, 2, 2, 2, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 2, 0, 1, 2, 2, 0, 1, 0, 2, 3, 0, 1, 0, 2, 0, 0, 2, 0, 1, 1, 0, 2, 0, 2, 2, 2,
1, 0, 1, 1, 0, 1, 1], [1, 1, 0, 0, 2, 0, 0, 1, 1, 1, 1, 0, 2, 2, 2, 2, 0, 0, 2, 0, 1, 0, 2, 2, 0, 1, 1, 0, 2, 0, 2, 2, 0, 1, 0, 0, 2, 2, 0, 3, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1], [0, 1, 1, 0, 2, 2, 2, 0, 1,
1, 0, 2, 0, 0, 0, 3, 2, 2, 0, 1, 1, 2, 0, 1, 1, 0, 0, 2, 0, 2, 0, 1, 1, 1, 1, 0, 2, 1, 0, 2, 2, 2, 0, 0, 1, 1, 0, 1, 1], [0, 0, 1, 0, 0, 0, 2, 0, 1, 1, 0, 2, 2, 0, 1, 1, 0, 2, 0, 0, 1, 0, 2, 0, 0, 1,
2, 0, 2, 0, 1, 0, 0, 0, 2, 2, 0, 2, 2, 0, 2, 0, 1, 1, 0, 1, 1, 0], [0, 1, 1, 1, 1, 0, 2, 2, 0, 1, 0, 2, 0, 0, 0, 1, 0, 2, 2, 2, 0, 1, 2, 2, 2, 0, 2, 2, 0, 2, 2, 0, 1, 0, 2, 2, 2, 0, 2, 2, 0, 2, 2, 0,
1, 1, 0, 1], [1, 1, 0, 0, 1, 1, 0, 2, 0, 0, 1, 2, 2, 0, 1, 1, 1, 0, 0, 2, 2, 0, 3, 0, 2, 0, 2, 0, 2, 2, 0, 1, 1, 0, 2, 0, 1, 0, 2, 0, 0, 2, 0, 1, 1, 0, 1, 0, 1, 1], [1, 0, 0, 1, 1, 0, 1, 2, 2, 2, 0, 0,
2, 0, 0, 1, 1, 1, 0, 2, 0, 0, 1, 2, 0, 2, 0, 2, 0, 0, 0, 1, 0, 2, 2, 0, 2, 2, 0, 2, 2, 0, 1, 0, 1, 1, 1, 1, 0], [1, 0, 1, 1, 0, 1, 0, 1, 0, 2, 2, 2, 0, 2, 2, 2, 0, 1, 0, 0, 2, 2, 2, 0, 2, 2, 2, 0, 2,
1, 1, 1, 0, 2, 2, 2, 0, 3, 2, 0, 1, 1, 1, 0, 1, 0, 1, 1], [0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 2, 2, 0, 0, 2, 0, 1, 1, 1, 0, 0, 2, 0, 1, 0, 3, 0, 2, 2, 2, 0, 1, 0, 1, 0, 0, 0, 3, 0, 2, 0, 1, 0, 1, 0, 1,
1], [1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 2, 0, 1, 2, 0, 1, 0, 1, 1, 0, 2, 2, 0, 3, 3, 0, 0, 0, 2, 0, 1, 0, 1, 0, 3, 3, 3, 0, 2, 2, 0, 1, 1, 1, 0, 1, 0, 1], [0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 2, 2, 0,
0, 1, 0, 1, 0, 1, 0, 2, 2, 0, 3, 3, 3, 0, 2, 0, 1, 2, 2, 2, 2, 0, 3, 0, 0, 2, 3, 0, 1, 0, 0, 1, 1, 0], [1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 2, 2, 2, 0, 0, 1, 1, 1, 1, 1, 0, 2, 0, 0, 3, 0, 1, 2, 2,
0, 0, 2, 2, 0, 2, 2, 2, 0, 1, 1, 1, 1, 0, 1, 0], [1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 2, 1, 0, 3, 3, 0, 0, 2, 0, 2, 1, 1, 0, 2, 2, 2, 0, 0, 3, 0, 0, 0, 0, 0, 1, 1],
 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 2, 2, 0, 3, 3, 0, 2, 2, 2, 0, 2, 0, 0, 3, 0, 0, 0, 1, 2, 2, 2, 2, 2, 2, 2, 0, 1], [0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1,
1, 0, 1, 0, 2, 0, 3, 0, 0, 0, 2, 0, 0, 3, 2, 2, 2, 2, 2, 2, 0, 0, 2, 0, 0, 0, 0, 0, 2, 0, 1], [0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 2, 0, 2, 2, 2, 2, 2, 0, 3, 0, 0, 0,
```

Functions

- get_colors: Defines the color maps that the maze will use
    - Returns: A dictionary that will correlate the integer key to a color
- save_maze: Saves a maze object to a file
    - Returns: None
- load_maze: Reads a maze from a png file into a 2d list with values corresponding to the known color
    - Returns: A maze as a list
- pos_chk: Validates if the coordinates specified (x and y) are within the maze
    - Returns: True if within the maze boundaries, False if not
- **generate_maze:** Generates a maze using the Depth First Search method
    - Returns: None
- solve_maze: Solves a maze using the Depth First Search method
    - Returns: None

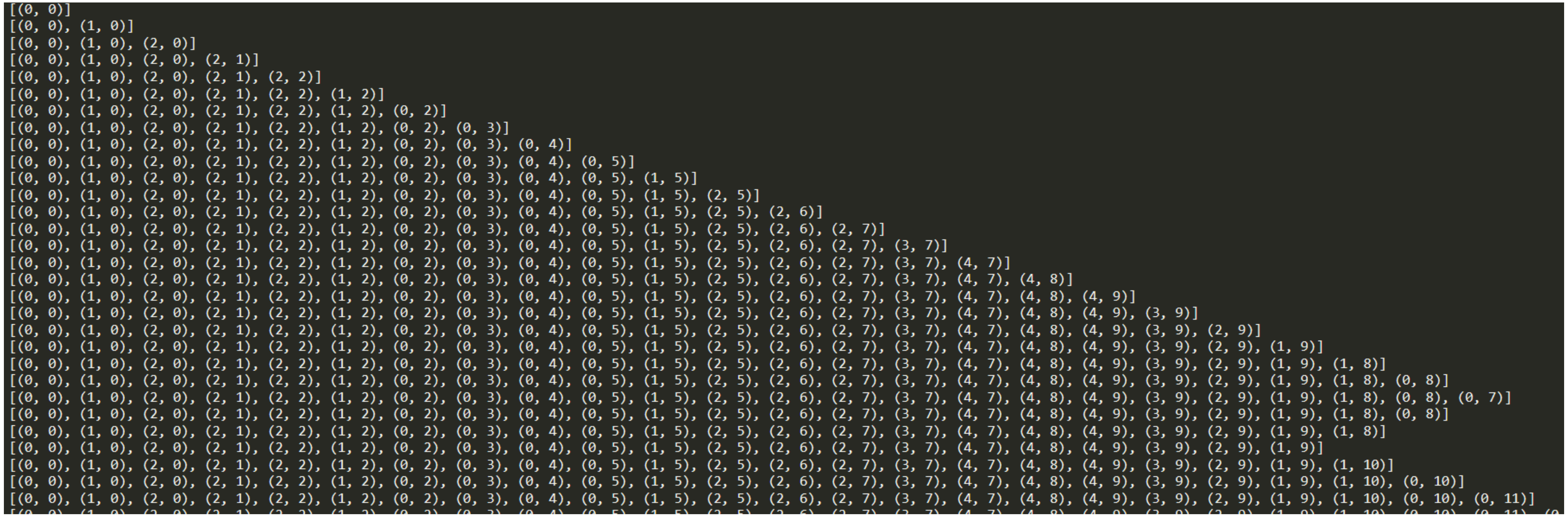
Generate_maze

- When generating the maze you are creating all the pathways for the maze **and you record each position explored in your stack!**
- Start at the [0, 0], End at [W/n_blocks, H/ n_blocks] of the image (W = H)
- Black (**key = 0**) indicates an unexplored path or coordinate, you are to explore with white – **key = 1 -> these are recorded in your 2d color list**
- **Movements:** Up, Down, Left, or Right
- **Condition:** You check the **neighbors and (neighbors of neighbors)** to see if there is an already explored path or coordinate in the stack
- **Selection:** Select randomly from pathways/coordinates that satisfied the above condition
- **Pop out:** You remove the most recent entry of the list if there are no moves to explore (blue path in video), the you repeat the exploration process (**Show whiteboard demo!)**
- **Completion:** Stop generating a maze when the stack is empty. In the previous video the blue path indicates that the path has been explored and lead to a dead end (popped each coordinate out of the stack). When you've explored every path – you keep popping until your list is empy.



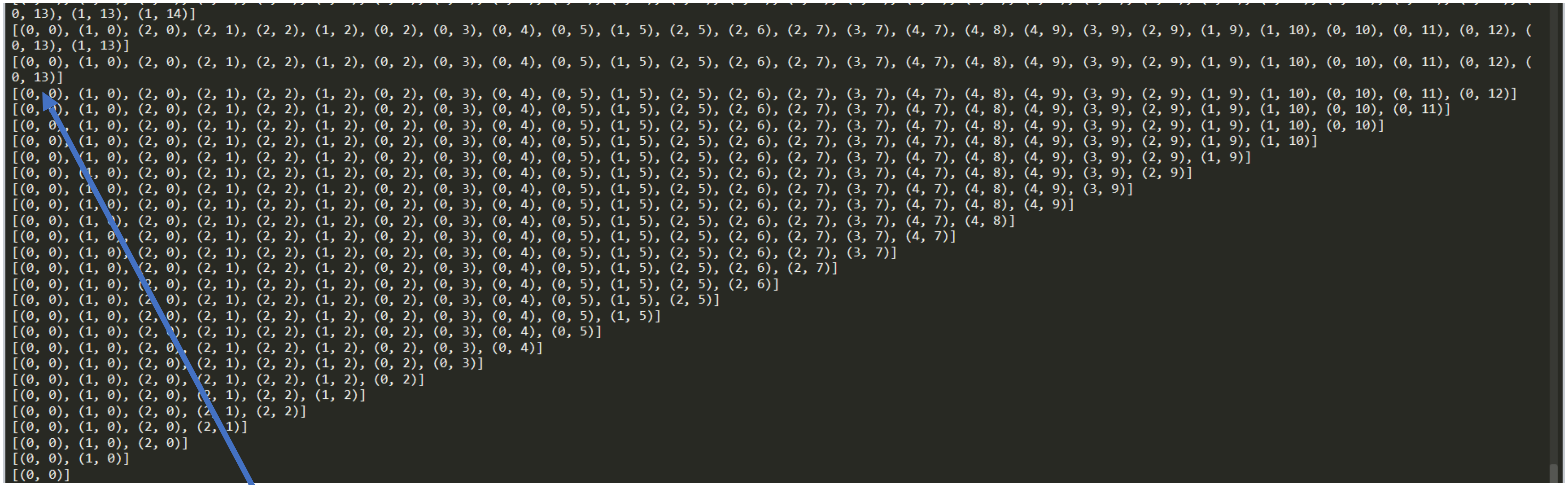
Stack close to the start of exploration...

Begin Maze Generation at [0, 0]



```
[(0, 0)]
[(0, 0), (1, 0)]
[(0, 0), (1, 0), (2, 0)]
[(0, 0), (1, 0), (2, 0), (2, 1)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9), (3, 9)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9), (3, 9), (2, 9)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9), (3, 9), (2, 9), (1, 9)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9), (3, 9), (2, 9), (1, 9), (1, 8)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9), (3, 9), (2, 9), (1, 9), (1, 8), (0, 8)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9), (3, 9), (2, 9), (1, 9), (1, 8), (0, 8), (0, 7)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9), (3, 9), (2, 9), (1, 9), (1, 8), (0, 8)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9), (3, 9), (2, 9), (1, 9), (1, 8)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9), (3, 9), (2, 9), (1, 9)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9), (3, 9), (2, 9), (1, 9), (1, 10)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9), (3, 9), (2, 9), (1, 9), (1, 10), (0, 10)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9), (3, 9), (2, 9), (1, 9), (1, 10), (0, 10), (0, 11)]
```

Stack towards the end...



```
0, 13), (1, 13), (1, 14)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9), (3, 9), (2, 9), (1, 9), (1, 10), (0, 10), (0, 11), (0, 12), (
0, 13), (1, 13)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9), (3, 9), (2, 9), (1, 9), (1, 10), (0, 10), (0, 11), (0, 12), (
0, 13)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9), (3, 9), (2, 9), (1, 9), (1, 10), (0, 10), (0, 11), (0, 12)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9), (3, 9), (2, 9), (1, 9), (1, 10), (0, 10), (0, 11)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9), (3, 9), (2, 9), (1, 9), (1, 10), (0, 10)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9), (3, 9), (2, 9), (1, 9), (1, 10)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9), (3, 9), (2, 9), (1, 9)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9), (3, 9), (2, 9)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9), (3, 9)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8), (4, 9)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7), (4, 8)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7), (4, 7)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7), (3, 7)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6), (2, 7)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5), (2, 6)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5), (2, 5)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5), (1, 5)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4), (0, 5)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3), (0, 4)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2), (0, 3)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2), (0, 2)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (1, 2)]
[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2)]
[(0, 0), (1, 0), (2, 0), (2, 1)]
[(0, 0), (1, 0), (2, 0)]
[(0, 0), (1, 0)]
[(0, 0)]
```

At this point you are popping every entry in the position stack out!

Try it yourself!

- Complete the following functions:
  - **generate_maze**:

- Next week:
  - **solve_maze** (This will be your WC7)
  - **Lazor Project Discussion**

Questions?