Maze submittal Documentation

- In order to submit your maze, you need to call the /submitMaze endpoint.
- Requirements of the submitted maze:
  - The maze needs to be a 2d numpy array.
  - The maze dimensions must be 10x10.
  - Each element in the numpy array should be either 1, 2, 4 or 8.
  - All the cells in the maze should be reachable, indicating that there can't be any blocked cells.
  - You must be able to provide your own solution for the maze.
  - Your maze will be evaluated against your agent before being allowed to solve other mazes.
  - In case of failure to submit a maze, your team will be disqualified from the final phase.
- To call the endpoint you need to send the following parameters:
  - agentId: Which is the agentId sent to you via email.
  - submittedMaze: Which is the maze you will submit for the final stage for the other contestants to solve. You need to convert the numpy array of the maze to a list then convert it to a json string to be able to call the endpoint correctly.
  - Below is a python example of how to submit a maze.

```
requests.post(f'{serverIp}/submitMaze',

json= {

  "agentId": "ABC123", "submittedMaze": json.dumps(maze.tolist())

} )
```

- How the maze is represented:
  - The maze is represented as a 2d numpy array.
  - Each entry in the numpy array is either 1, 2, 4 or 8.
  - The 2d array is transposed (rows are changed into columns and columns into rows)
  - If the cell value is 1 it indicates a north exit, 2 indicates an east exit, 4 indicates a south exit and 8 indicates a west exit.
  - To understand the logic behind this, try to think of 1,2,4 and 8 as bits indicating **WSEN [“West”, “South”, “East”, “North”],** where the bit with a value of 1 indicates an exit in that direction. For example, 1: 0001, the last bit represents North, indicating a North exit. 2: 0010, the third bit represents East, indicating an East exit. 4:0100, the second bit represents South, indicating a South exit. 8: 1000, the first bit represents West, indicating a West exit.

Example maze:

Maze array:

[[2 4 2 1 1 1 1 4 4 2]

[4 8 2 2 1 1 8 2 1 1]

[2 1 1 4 2 8 8 4 4 2]

[4 8 2 8 4 2 8 1 1 2]

[4 2 1 8 2 4 4 2 8 2]

[8 2 4 8 4 4 2 4 8 2]

[8 2 8 4 2 1 1 1 1 2]

[8 4 8 8 2 2 4 4 8 2]

[8 1 2 1 2 2 8 2 1 1]

[4 8 1 8 1 4 8 1 4 8]]

Firstly, the maze is transposed:

[[2 4 2 4 4 8 8 8 8 4]

[4 **8** 1 8 2 2 2 4 1 8]

[2 2 1 1 4 8 8 2 1]

[1 2 4 8 8 8 4 8 1 8]

[1 1 2 4 2 4 2 2 2 1]

[1 1 8 2 4 4 1 2 2 4]

[1 8 8 8 4 2 1 4 8 8]

[4 2 4 1 2 4 1 4 2 1]

[4 1 4 1 8 8 1 8 1 4]

[2 1 2 2 2 2 2 2 1 8]]

Remember: 1: North exit, 2: East exit, 4: South exit, 8: West exit.

To get an idea of how each cell is represented, let's look at the cell highlighted in green. The value of the cell is 8, indicating that it has a West exit, the cell to its north has value of 4, which indicates a south exit, which traces back to our cell. So all in all, the cell has both a west exit and a north exit.