**Describe the state of your project, what works and what doesn’t.**

I have implemented the modifications to the solve and getPath methods to handle branching paths and backtracking in the CaveExplorer class. The code appears to be logically sound and is working as expected.

**Describe how you tested your program, including tests that made you rethink your code. Include the layouts you used.**

I have tested the program using various layouts to verify its functionality and correctness. Here are some of the layouts I used for testing:

5 6

RRRRRR

R..S.R

R.R.RR

R.MRRR

RRRRRR

5 6

RRRRRR

RS...R

R.R.RR

R.RMRR

RRRRRR

5 6

RRRRRR

R.R.RR

R..S.R

R.MRRR

RRRRRR

I also tested layouts with dead ends and different possible paths to the mirror pool to ensure that the program handles branching paths and backtracking correctly.

**In a sentence or two, what did you learn?**

I learned about implementing depth-first search (DFS) with backtracking to solve problems efficiently. Additionally, I improved my understanding of how to work with stacks and visited arrays in algorithmic problem-solving.

**In a sentence or two, what did you like about this project?**

I liked that this project provided an opportunity to apply algorithmic thinking to solve a cave navigation problem. It allowed for the implementation of both the solve method and the getPath method, demonstrating how these components can work together to solve a real-world problem.

**In a sentence or two, what did you find confusing or would like to see done differently regarding this project?**

Initially finding a path by backtracking was confusing. But later, I figured out to use stack to find the path without getting dead end.

**In a sentence or two, if you had another hour or two, what would you like to add to the project or how would you do things differently?**

If I had additional time, I might consider adding feature that generates random cave layouts, increasing the variety and complexity of the mazes that the cave explorer can navigate, thereby adding an element of unpredictability and challenge to the program.