CS2040S: Data Structures and Algorithms

Problem Set 7 Part 2

Due: Tuesday, March 30, 11:59pm

Collaboration Policy. You are encouraged to work with other students on solving these problems. However, you **must** write up your solution **by yourself**. We may, randomly, ask you questions about your solution, and if you cannot answer them, we will assume you have cheated. In addition, when you write up your solution, you **must** list the names of every collaborator, that is, every other person that you talked to about the problem (even if you only discussed it briefly). You can do so by leaving a comment at the start of your .java file. Any deviation from this policy will be considered cheating, and will be punished severely, including referral to the NUS Board of Discipline.

Overview

This is a continuation of the previous problem set.

Your second job involves determining how best to use your superpower. You have been given the power to bypass some fixed number of walls. For example:

- Able to walk through walls.
- Able to fly over walls.
- Strong enough to knock down walls.
- Possesses dynamite.
- Some other superpower.

For zero points of extra credit, explain how your superpowers work and how you acquired them (e.g., on the forum). In any case, your job is to find the shortest path from the source to the destination, while bypassing no more than the allowed number of walls.

Problem Details

Problem 7.d. Maze Exploration for Real SuperPeople

In this part, as in the previous part, your job is to determine the shortest path from a specified source to a specified destination. However, you are also given the ability to bypass (demolish, jump over, or magically traverse) up to a limited number of walls along the way. Of course, you will not be allowed to use your power to demolish the outer walls that surround your maze. You begin with a fixed amount of superpowers, and every time you demolish a wall, your power reduces by one. Your goal is to find the shortest path from the start to the end.

Create a new class `MazeSolverWithPower` that implements `IMazeSolverWithPower` which contains the overridden method:

```
Integer pathSearch(int startRow, int startCol, int endRow, int endCol, int superpowers).
```

It should return an integer representing the minimum number of steps needed if a path is found, or `null` if no such path is available. As before, it should also update, for each room, the `onPath` attribute. (Notice that the `IMazeSolverWithPower` interface inherits from `IMazeSolver`, and hence must correctly implement all the methods from there as well. In this case, if `numReachable` is called after a `pathSearch` with superpowers, then its answer should also be based on having the same number of superpowers (in addition to having the same starting location))

Hint: One strategy is to represent the state as a combination of the current room and the remaining amount of superpower. If you explore this graph of all possible states, you will visit all possible rooms, with all possible remaining amounts of superpower.