etraveli group

MAZE CHALLENGE

OVERVIEW

The Maze Challenge is a part of the Etraveli Group hiring process, and will help us determine your level of competence. You are strongly encouraged to approach it with a professional attitude and deliver a working, production - grade solution. Generally speaking, we have found that most candidates need to devote a minimum of 8 hours on the challenge in order to create a solid submission. Please keep in mind that this code will be reviewed and evaluated, and we expect you to be able to present it and possibly even change it during your interview.

Below you will find the requirements of the challenge in full detail, as well as a comprehensive QA section, which you are urged to consult before contacting us with any questions. We will of course be available for any additional clarifications or questions that may arise. As a final note, we wish to thank you in advance for your time and effort, and wish you good luck.

REQUIREMENTS

The problem

Given a two - dimensional matrix representing a maze with a single arbitrary starting and ending point, as well as an arbitrary number of wall blocks, you are required to produce an application that will output an actor’s route in the form of two - dimensional points from start to finish. The actor cannot possess any knowledge of the route beforehand; it is required to “discover” the route progressively, being only able to inspect and move to any squares that are North, South, East and West from its current position.

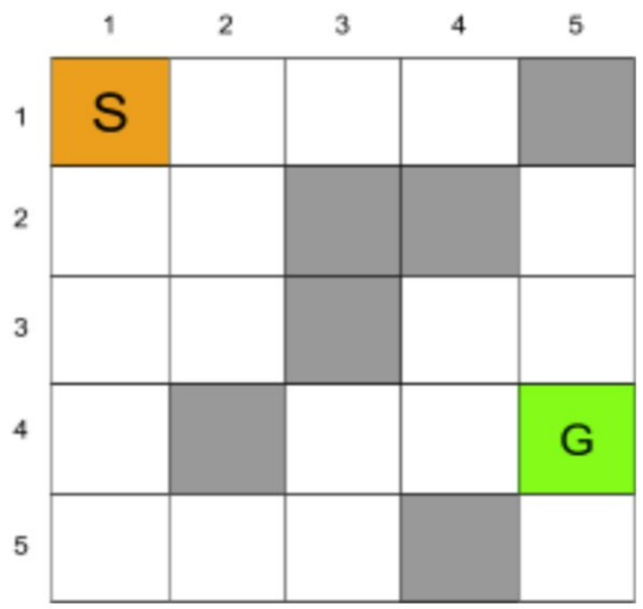

Figure 1: A sample maze

(1:1 (S)), (2:1), (3:1), (4:1), (5:1), (5:2), (5:3), (4:3), (4:4), (4:5 (G))

Figure 2: Example output

Required deliverable traits

- Demonstrable OO design principles. We want the code to be as clearly written and partitioned as you can make it.
- The implementation of the solver should be written in Java Programming language.
- Comprehensive unit tests.
- The application must support maze definitions read from an external resource of your choice, like a file or the command line. The format of the maze definition is up to you, but it has to be an external input source. Solutions in which any part of the maze is hardcoded into the code are not acceptable.
- The submission is considered invalid if it does not include the full source code, as well as comments where necessary.
- In case you use external sources, libraries or research, be sure to include your sources in a References section at the bottom of the document.

Bonus points

- Making it easy for the reviewer to see how the algorithm works.
- Supporting multiple path finding algorithms.
- Technical concerns: Logging, Caching, runtime constraints and parameters validation, etc.

QA

Q: How big can the maze be?

A: Anywhere from 2 to Integer.MAX_VALUE, although you should try to keep things reasonable. We are after a demonstration of your engineering and coding skills, not the most efficient maze solving algorithm ever.

Q: Should the Starting or Ending point be at the edges of the maze?

A: No. It can be any block in the two - dimensional matrix, although the two should not coincide.

Q: What is the preferred input format for the application?

A: Although any form of input is acceptable, you along with us would be probably better served with an easy to read format, like a text file with contents similar to the following example:

```
____G__X

___XXX__

X______X

__XXXX__

___X____

__S__X__
```

Figure 3: Sample maze definition

Q: I discovered a bug but have already submitted my code! What can I do?

A: Tell us! It is ok to re-submit your code within a reasonable amount of time.