

Programming Assignment 5

Due at your recitation session on September 30-October 4, 2013

Reading

Read Sections 7 (preamble), 7.1, 7.2, 7.3, 7.5, and 7.6 in the textbook.

Programming

First, make any changes that were discussed in your recitation session and refactor your code in light of the principles of routine design.

Then, implement the following additional `Maze` methods:

- `MazeRoute routeGreedy(MazeCell initialCell)` returns a route within the maze starting from and including the given initial cell assuming that in each cell the mouse chooses the passage with the smallest travel time. The mouse should never cross a passage whose travel time is `MAX_VALUE`. The route should stop if the mouse exits the maze, reaches a dead end, or reaches a cell which it had previously visited. It returns an empty list in case of error, for example if at any point the mouse finds itself in a cell that does not belong to the maze (either at the initial cell or throughout the route). The method throws a `UninitializedObjectException` if this `Maze` is invalid.

The three methods `routeFirst`, `routeRandom`, and `routeGreedy` follow the same general pattern, with the only difference being the specific hop selection. The three route methods can then be generalized and simplified with a *function object*. Define the interface `PassageSelector` with a single method `MazeCell nextCell(MazeCell currentCell)` that chooses the next cell. The `PassageSelector` will be implemented in three different ways: select an arbitrary next cell, select a random next cell, and select a passage with the smallest travel time. (As usual, the mouse should never cross a passage whose travel time is `MAX_VALUE`.) If no feasible passage exists, `nextCell` will return `null`. A `PassageSelector` object is called a function object because it is an object that wraps the invocation of the `nextCell` method.

Then, implement the following `Maze` methods:

- `MazeRoute route(MazeCell initialCell, PassageSelector passageSelector)` returns a route within the maze starting from and including the given initial cell assuming that in each cell the mouse chooses the passage with the given passage selector. The mouse should never cross a passage whose travel time is `MAX_VALUE`. The route should stop if the mouse exits the maze, reaches a dead end, reaches a cell which it had previously visited, or the `PassageSelector` selects no next cell. It returns an empty list in case of error, for example if at any point the mouse finds itself in a cell that does not belong to the maze (either at the initial cell or throughout the route). The method throws a `UninitializedObjectException` if this `Maze` is invalid.
- `Double averageExitTime(MazeCell outside, PassageSelector passageSelector)` returns the average time for a mouse to reach the outside cell if the mouse starts from a random initial cell (other than outside). The exit time should be evaluated by averaging the route time starting from every initial cell in the maze (except the outside cell). If the route time from any initial cell is `MAX_VALUE`, then the average exit time should also be `MAX_VALUE`. The method throws a `UninitializedObjectException` if this `Maze` is invalid.

As usual, your classes may contain as many auxiliary private methods as you see fit, and additional helper classes may be defined. Your classes must be documented in JavaDoc. Test cases must be created in JUnit to test all its public methods.

For extra credit, visualize the maze using JGraph or similar libraries.

Discussion Guidelines

The class discussion will focus on routines.

On-Line Submission

Bring a copy to recitation to display on a projector. Additionally, submit an electronic copy of your program to blackboard. Turn in with your code:

- A README file explaining how to compile and run the code.
- A Make, Ant, or Maven build file.
- A comment at the top over file containing your name, email address, a one-sentence description of the file, and if necessary a longer comment describing the design of the file.

The code should be handed in a zip, tar.bz2, or tar.gz archive. Archives in 7z cannot be accepted.