Data Structures Assignment 4:
Rat's Nest

Instructor: Benjamin Geiger (`begeiger@mail.usf.edu`)

Assigned 10 July 2014 — Due 17 June 2014, 9:30 AM

Assignment

Download the `RatsNestProject.tar.gz` file from Canvas and extract it. You will find several source files.

Two methods in `Maze.cpp` are incomplete: `findExitPath()` and `findShortestPath()`. You are to implement these methods, using depth-first search and breadth-first search, respectively.

Details

The `Maze` class implements a maze, represented as a square grid of rooms. Two 'open' rooms that are 4-adjacent (adjacent horizontally or vertically, but not diagonally) are considered connected. Your code should find paths from the start to the exit.

Note: When searching for an open room, you must search neighbors in the following order: down, left, up, right. For your convenience, the `direction` enum can be incremented in this order:

```
for (direction d = DOWN; d != NONE; d++) { ... }
```

Compilation and Testing

To compile your program in Linux, simply type `make` at the command prompt. The included `Makefile` should handle the rest.

A sample input file (`input0`) and output file (`correct0`) have been provided. To test your code:

1. Compile your code, as above.
2. Enter at the command prompt:
   ```
   ./RatsNest < inputBFS > outputBFS
   ```
   This runs the program and tells it to accept input from `input0` and save output to `test0`.

   (*Mutatis mutandis* for DFS.)

3. Compare the output to the correct output:

   ```
   diff -u outputBFS correctBFS
   ```

   Lines where your output differs from the correct output will be printed; your lines will be prefaced with - and the correct lines will be prefaced with +. (Lines with no characters are common to both and are included for context.)

Administrivia

- Include your name in the comment at the beginning of the `Maze.cpp` file.
- Submit only `Maze.cpp`. The other files should not be modified.
- Your code will be tested on a Linux system; other Unix systems are likely to work properly, but Windows (specifically Visual Studio) often causes problems. If in doubt, test your code on the C4 lab PCs or on CIRCE.

  Submissions that do not compile will be given a grade of 20%.
- Your code will be tested on several input files other than those provided.
- Please properly indent your code. Most code editors contain an automatic indentation feature.