Board Design

Overview

Finding an optimal solution to a maze is a very challenging problem in computer science. Even with the fastest machines, there are times where solutions must be approximated, rather than calculated exactly (especially for complex boards). Luckily, this is a problem for everyone, and you may not need to achieve perfection to get the treasure first!

To make this project easier, we've designed the boards in several levels of increasing difficulty. Both the training boards that we've provided for you, and the test boards that we will use to test your algorithm will have at least some boards at each of these levels.

Each board has a limited number of steps. We set these limits so that is possible to obtain all the treasure with plenty of extra steps, however you may have to be very clever on some boards to reach the treasure without exceeding the limit.

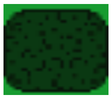
Terrain Types

Before we jump into the levels of boards, it is best to review the different types of terrain the boards may contain and their respective step cost to step onto the square.

Open Space – 1 step

Mud – 5 steps

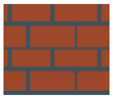
Water – 13 steps


Bushes – 23 steps

Trees – 37 steps

Mountain – 47 steps

Lava – 61 steps


Wall – All steps

Regular Boards

Level 0

These are open boards with a single treasure. To succeed, you need only make moves in the direction of the treasure.

These boards are extremely easy for humans to solve, but they are more complex than they appear at first. There are a lot of possible moves that might be made at each step. Analyzing all combinations and finding the optimal path can become too expensive for a computer, even for smaller boards. Some "advanced" solutions might need some creativity.

Feature	Value
Number of treasures	1
Type of (non-perimeter) obstacles	None
Open path exists	Yes
Open path is optimal	Yes
Optimal solution has moves away from treasure	No
Dead ends	No

Level 1

Level 1 boards add non-wall obstacles. Stepping on these obstacles will reduce your score much more than stepping on an open space. An optimal path that does not require stepping on an obstacle will always exist. Further more all steps on this solution will be in the direction of the treasure. There are no dead ends on these boards.

A simple algorithm for improving your score on these boards might be to choose a non-obstacle move in the direction of the treasure.

Feature	Value
Number of treasures	1
Type of (non-perimeter) obstacles	Any, except Walls
Open path exists	Yes
Open path is optimal	Yes
Optimal solution has moves away from treasure	No
Dead ends	No

Level 2

Level 2 boards add some dead ends. This means that (at any point) there may be two paths in the direction of the treasure, one of which has a dead end.

There are a couple of options here:

1. Handle the dead end by going through the obstacle to get to the treasure. (This is not optimal, but it will get you there).
2. Backtrack your steps until you reach a square that has an open & unexplored path in the direction of the treasure and explore that path.
3. Backtracking is still not quite optimal, since you've wasted steps going down a road with a dead end. The best solutions will analyze the board ahead of time and choose a path that avoids dead ends all together. This approach is addressed in college level algorithm courses, and we don't expect you to understand it now. However, if you are interested, these are some algorithms to check out:
    - https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm
    - https://en.wikipedia.org/wiki/Depth-first_search

Because all solutions are in the direction of the treasure, you have at most two moves to consider each time a step is made.

Feature	Value
Number of treasures	1
Type of (non-perimeter) obstacles	Any, except Walls
Open path exists	Yes
Open path is optimal	Yes
Optimal solution has moves away from treasure	No
Dead ends	Yes

Level 3

These boards build on level 2 by allowing the solution path to temporarily move away from the treasure. This means that in general you cannot assume that the best move is toward the treasure at this level and above.

This is a true maze-solver solution, which is a very hard problem. When you have great solutions here you have risen to the challenge and should be proud of your accomplishments.

Feature	Value
Number of treasures	1
Type of (non-perimeter) obstacles	Any, except Walls
Open path exists	Yes
Open path is optimal	Yes
Optimal solution has moves away from treasure	Yes
Dead ends	Yes

Bonus Boards

Level 3 is a hard problem and starting with level 4 we've made it (much) more difficult. Optimal solutions are extremely difficult to calculate. (Most of the volunteers can't do it with the resources that you have).

So, with these boards, you want to get a solution that is as close to optimal as possible, and maybe be happy with any reasonable solution.

Level 4

Level 4 has many treasures on an open board. This seems trivial, but it quickly becomes complicated when you think about how to go from treasure to treasure in the most efficient way. This is a very hard problem that you don't have to solve...you just have to figure out how to be better than the other schools.

Feature	Value
Number of treasures	Many
Type of (non-perimeter) obstacles	None
Open path exists	Yes
Open path is optimal	Yes
Dead ends	Yes

Level 5

Level 5 extends level 4 by adding some obstacles. There is a path through the maze that secures all the treasure, finding it is not so easy.

Feature	Value
Number of treasures	Many
Type of (non-perimeter) obstacles	Any, except Walls
Open path exists	Yes
Open path is optimal	Yes
Dead ends	Yes

Level 6

Level 6 uses walls for all the obstacles. To get the treasure, you must find an open path through the maze. Stepping on a wall will consume all your remaining steps.

Feature	Value
Number of treasures	Many
Type of (non-perimeter) obstacles	Walls
Open path exists	Yes
Open path is optimal	Yes
Dead ends	Yes

Level 7

Up until level 7, the best solutions avoided obstacles. At this level, the best solution may be in fact to step on an obstacle. An open path will always exist to the treasures, however the cost of following it may be greater than going through the obstacle.

This is a VERY hard AI problem in general. Exploring all possible combinations may be the only way, but this will take more CPU time than we give your algorithm to run. The best approach might be to consider a couple of candidate solutions, and pick the best on -- this approach is still not trivial.

Feature	Value
Number of treasures	Many
Type of (non-perimeter) obstacles	Any, except Walls
Open path exists	Yes
Open path is optimal	No
Dead ends	Yes

Level 8

Level 8 is the final level. At this level, the only way to get to the treasure is to go through an obstacle. This is a good test of what your path finding algorithm does. If it considers only non-obstacle paths, what does it do if there is no such path.

A simple fall back of "go straight for the treasure(s) if I can't find an open path" might pay big dividends at this level.

Feature	Value
Number of treasures	Many
Type of (non-perimeter) obstacles	Any, except Walls
Open path is optimal	No
Dead ends	Yes

Making Boards

This material is completely optional but might be useful for debugging or improving your solution.

Description

Each board is technically a text file in comma separated values (CSV) format. Spreadsheet-like data is often in this format and many spreadsheet and other programs can read and update the data (e.g. Microsoft Excel, Numbers on Mac). You do not need to open the file using a fancy spreadsheet editor, however. You can open these using any text editor like VS Code, Notepad, even DrJava.

First, open the board `Level_0_0.csv` in a text editor. This is what it should look like:

```
#,#,#,#,#,#,#,#,#,#,#,#
#, , , , , , , , , ,T,#
#, , , , , , , , , , ,#
#, , , , , , , , , , ,#
#, , , , , , , , , , ,#
#, , , , , , , , , , ,#
#, , , , , , , , , , ,#
#, , , , , , , , , , ,#
#, , , , , , , , , , ,#
#,P, , , , , , , , , ,#
#,#,#,#,#,#,#,#,#,#,#,#
1000,,,,,,,,,,
```

`Level_0_0.csv` is an 11 row by 12 column (11x12) board. Each 'column' of the board must be separated by a comma, so each row will contain have 12 – 1 = 11 commas. The symbols are as follows:

- `#` are walls. The game will ensure that the board is surrounded by a complete wall. There is no escape!
- A single blank character is an open space.
- `T` is a treasure. More than one treasure can appear in a board.
- `P` is the starting position of the player.
- `,` commas are necessary to separate the value for each column.
- `1000` is the maximum number of steps for the board. You may set this value.
- The last set of commas after 1000 is optional (our boards might be missing commas).

Get Started

Open a provided board in a text editor or spreadsheet program (which better show the rows and columns). Be sure to 'Save As' immediately and create a new name for the board – we don't want you to edit the provided boards!

Be sure to save the file as a **.csv** and not as a .xls or .ods if you're using a spreadsheet program.

When you're ready to go, use the following codes to place different types of terrain or treasures:

Symbol	Description
P	Player starting position (only put one)
T	Treasure
A blank space	Open Space
#	Wall
m	Mud
w	Water
B	Bushes
t	Trees
M	Mountain
L	Lava