Challenge Problem 2
Can ‘U’ Escape?

By: Alex Li
ACM-ICPC Club

Problem Description

https://github.com/OSUACM/Weekly_Events/blob/master/180217.md

Assumptions:

Must contain exactly 1 U

Must be a rectangular maze

‘Edge’ cases

Always make sure to consider these before you consider your solution to be working - either at start or at the end

1. U is actually on the edge

**
*U

2. U is alone

U

Simple, but how to do it?

- “Implementation Problem”
- Goal: Solve it as simply as possible

My Solution

Ind	0	1	2	3
0	*	*	*	*
1	*	U	.	.
2	*	.	.	*
3	*	*	*	*

Make a Queue of positions to be reached. At first, just add the position of U to the queue.

While the queue is not empty, get the next element of the queue.

> Add all 4 squares next to it to the queue (above, below, left, right), unless those squares correspond to walls of the maze
>
> If any element is at the edge of the maze, there must be an exit.

Make it smarter!

- Boolean[][] to store seenElements
- Initially true only for U’s position
- Set it to true each time we find a new element.
- Do not add elements to positions that are set to true.

The Code

Java

Slightly more Concise

Generalizations to other problems

- With graph theory, keeping an array of seen elements is a very common technique
- Using the described algorithm, we can search systematically through multidimensional regions (though there will be a lot of if statements for higher dimensions)
- Using classes can be helpful

Thanks for listening!

Questions?

Comments?