

CS2134 Maze Spring 2016

4/17/2016

Written Part: Extra Credit

1) The difference between my recursive method and my queue method is that one checks one branch at a time, and one checks them simultaneously.

Recursive

```
XXXXXXXXXXXXXXXXXXXXX
s0x000x0000x000exx
x0x0x0x0xx0x00xxxx
x000x000xx0x00x.xx
xxx.x00xxx000xx.xx
xxx.xx..xx00...xx
xxx...x.xxxxxxxxxxx
xxxxx.x.....xxx
xxxxxxxxxxxxxxxxxxxxx
```

Queue

```
XXXXXXXXXXXXXXXXXXXXX
s0x000x0000x000exx
x0x0x0x0xx0x00xxxx
x000x000xx0x00x.xx
xxx0x00xxx000xx0xx
xxx0xx00xx000000xx
xxx000x0xxxxxxxxxxx
xxxxx0x00000000xxx
xxxxxxxxxxxxxxxxxxxxx
```

Recursive

```
XXXXXXXXXXXXXXXXXXXXX
s0x000x0000x0000xx
x0x0x0x0xx0x00xxxx
x000x000xx0x00x0xx
xxxex00xxx000xx0xx
xxx.xx00xx000000xx
xxx...x0xxxxxxxxxxx
xxxxx.x00000000xxx
xxxxxxxxxxxxxxxxxxxxx
```

Queue

```
XXXXXXXXXXXXXXXXXXXXX
```

```

s0x...x...x...xx
x0x0x.x.xx.x..xxxx
x000x...xx.x..x.xx
xxxex..xxx...xx.xx
xxx.xx..xx.....xx
xxx...x.xxxxxxxxxxx
xxxxx.x.....xxx
xxxxxxxxxxxxxxxxxxx

```

The recursive method is more effective for farther exits towards the top right corner, as it checks those branches first. So in the first maze example, the recursive function checks fewer places than the queue. In the second example, as the exit is closer to the start place, no matter which path it lies down, the queue will find it quickly. Whereas in the second example, the recursive function has to check the whole top right of the maze first.

2) The recursive method and the stack method take the same approach to the problem, in that they check branches and back track to the last split point. Hence they will visit the same spaces.