

The algorithm is given a maze (formatted as outlined in the assignment) and a starting (current) position  $p$ .

1. If  $p$  is an exit, mark it.
2. If  $p$  is a wall or has already been visited, return.
3. Get all positions neighboring  $p$  and recursively call the algorithm with the obtained positions.

The first eight steps are thus

```

0123456789A
0 #####
1 #  #  #  #
2 #  #  #  #  #
3 #   ## ## #
4 ###      #
5 #S  #  # ###
6 # ##  #  #
7 #   # ### #
8 #  #  #  #  #
9 #   #      #
A #####

```

| Depth | $p$   | obstacle/visited | neighbours              |
|-------|-------|------------------|-------------------------|
| 0     | (5,1) | no               | (5,0) (5,2) (4,1) (6,1) |
| 1     | (5,0) | yes              |                         |
| 1     | (5,2) | no               | (5,1) (5,3) (4,2) (6,2) |
| 2     | (5,1) | yes              |                         |
| 2     | (5,3) | no               | (5,2) (5,4) (4,3) (6,3) |
| 3     | (5,2) | yes              |                         |
| 3     | (5,4) | yes              |                         |
| 3     | (4,3) | no               | (4,2) (4,4) (3,3) (5,3) |