Problem 2: Stack-based

* 3,5
* 3,6
* 3,4
* 2,4
* 1,4
* 1,3
* 1,2
* 1,1
* 2,1
* 3,3
* 4,5
* 5,5

Problem 4: Queue-based

* 3,5
* 4,5
* 3,4
* 3,6
* 5,5
* 3,3
* 2,4
* 6,5
* 5,4
* 1,4
* 7,5
* 5,3

The stack-based algorithm used in problem 2 solved the maze with a depth-first search, meaning it searches in the same direction until it hits a dead end. The algorithm inspects the entirety of a path before moving on. The queue-based algorithm used in problem 4 solved the maze with a breadth-first search, meaning it searches starting with the squares closest to the start point. The order of squares searched ripples out from the origin. They differ in this way because stacks process the last item pushed onto it first (i.e. the last square of the path being explored), whereas queues store the last item at the end of the data structure (the first square not yet explored).