Avery Einhorn

UID: 405 115 287

CS32 S19 HW2

**2.** (3,5) (3,6) (3,4) (2,4) (1,4) (1,3) (1,2) (1,1) (2,1) (3,3) (4,5) (5,5)

1 2 3 4 5 6 7 8 9 10 11 12

**4.** (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 queue based implementation necessitates that items are added onto the back of the queue, and examined at the front of the queue, whereas the stack was added onto the top of the stack *and* examined at the top of the stack.

For the stack, that means that the *most recent*  item to be added is the *first* to be examined, so the algorithm essentially explored in one direction until it can’t go in that direction anymore, and then returned to the last known junction of two paths, to explore that other path.

For the queue, the *oldest* item in the queue is the one examined. Essentially, the queue implementation will examine a cell’s neighbors, if legal, it adds them into the queue, but then goes back and examines older stuff in the queue (could be all the way across the maze) before actually picking up on that path it just examined.