Problem 2:

The first 11(r, c) coordinates popped off are (6, 4), (6, 5), (7, 5), (8, 5), (8, 6), (8, 7), (8, 8), (7, 8), (6, 6), (5, 4), (4, 4)

Problem 4:

The first 11(r, c) coordinates popped off are (6, 4), (5, 4), (6, 5), (4, 4), (6, 6), (7, 5), (3, 4), (4, 5), (8, 5), (2, 4), (4, 6)

While using a stack, I directly accessed the element that I just pushed on and operated it. As for a queue, I could only access the front element of the queue but push elements onto the last position. Therefore, I could not continuously test whether certain path exists. Particularly, for the special case, if there is only one possible situation for each step, stack and queue behave similarly. Here is an instance for a typical case. I have four choices to leave (r, c). For a stack, I would pop (r, c) off the stack and push (r – 1, c), (r, c + 1), (r – 1, c), (r, c – 1) on. Then, I would first consider that I go west and stand on (r, c – 1) now. This time, I would still find any valid moving possibilities and continue similar operations until I stand on the destination or there is no way to go. For a queue, I would first pop (r, c) off the queue and push (r – 1, c), (r, c + 1), (r + 1, c), (r, c – 1) on. Then I would consider that I go north. But this time, if I find any valid cells that I could leave (r – 1, c), I would push these valid cells onto the queue and go back to (r, c + 1) to find any valid cells that I could leave (r, c + 1).