Problem 2:

(row, col)

1. (6, 4)
2. (6, 3)
3. (6, 5)
4. (7, 5)
5. (8, 5)
6. (8, 6)
7. (8, 7)
8. (8, 8)
9. (7, 8)
10. (6, 6)
11. (5, 4)
12. (4, 4)

Problem 4:

(row, col)

1. (6, 4)
2. (5, 4)
3. (6, 5)
4. (6, 3)
5. (4, 4)
6. (6, 6)
7. (7, 5)
8. (3, 4)
9. (4, 5)
10. (8, 5)
11. (2, 4)
12. (4, 6)

The two algorithms differ from each other since the top of the stack was the last coordinate to be pushed, and the front of the queue was the first coordinate that was pushed. This is because stacks are depth-first (follow a path all the way through) whereas queues are breadth-first (check closest coordinates first). In this problem, that means that stacks follow paths until a dead end or reaching the end coordinate, while queues check all the immediate adjacent spaces first, then the adjacent spaces after that, expanding in all directions until it reaches the end coordinate.