Problem 2:

1. (4, 3)

2. (3, 3)

3. (5, 3)

4. (5, 2)

5. (5, 1)

6. (6, 1)

7. (7, 1)

8. (8, 1)

9. (8, 2)

10. (6, 3)

11. (4, 4)

12. (4, 5)

Problem 4:

1. (4, 3)

2. (4, 4)

3. (5, 3)

4. (3, 3)

5. (4, 5)

6. (6, 3)

7. (5, 2)

8. (4, 6)

9. (5, 5)

10. (5, 1)

11. (4, 7)

12. (6, 5)

For Problem 2, a stack (first in last out) is used to store coords to be discovered. In this way, the last coord pushed in (A) will be first popped out and its neighbors will be discovered BEFORE any of the earlier pushed coords than A. It works the same for the neighbors of A. In other words, the computer will try to search a route as deep as possible unless a dead end is encountered and then it will start searching a new one.

For Problem 4, a queue (first in first out) is used to store coords to be visited. Thus, for each coord B, its neighbors will be discovered AFTER all the remaining coords in the queue when B is the top to be popped out. In other words, the computer tries to search different possibilities as broad as possible.