2.

6,4

6,3

6,5

7,5

8,5

8,6

8,7

8,8

7,8

6,6

5,4

4,4

4.

6,4

5,4

6,5

6,3

4,4

6,6

7,5

3,4

4,5

8,5

2,4

4,6

The stack algorithm uses a “depth-first search” (where it follows each potential path as far down as possible (until the path hits a dead end, like a “person running down a path until they hit a wall”), before it starts down another path) because each time it selects its next step, it searches around the latest position that it stored. Because of this, it will continue down the same path until it hits a dead end and must go back to the original place where the maze branched out.

The queue algorithm uses a “breadth-first search” (where it explores all potential paths pseudo-simultaneously in that it advances one step in all potential paths before advancing the next step down all potential paths, appearing to “ripple out” through all potential paths in the maze). This happens because each time the queue selects its next step, it searches around the first position that it stores. Because of this, a queue will find the optimal path through a maze; it never “puts all of its eggs in one basket,” so to speak.