Angela Kan

HW 2

#2)

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)

#4)

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)

Stacks follow the order of last-in-first-out. Therefore, the algorithm that uses the stack data structure pops the Coord at the top of the stack first (aka the last pushed item). Because the surrounding path coordinates are pushed in order of North->East->South->West, this means that the coordinates will generally be popped and checked in the opposite order (with West coming first, followed by South, etc). Therefore, more of the Southern part of the maze is checked with the stack algorithm in the beginning.

On the other hand, queues follow the order of first-in-first-out. Therefore, the algorithm that uses the queue data structure dequeues the Coord at the front of the stack first (aka the first enqueued item) and makes its way from front to back, adding any surrounding path coordinates to the back of the queue. This means that each enqueued coordinate will be checked and dequeued in the order it was added to the queue (which is usually in order of N->E->S->W).