2. (4, 6), (3, 6), (5, 6), (5, 7), (5, 8), (6, 8), (7, 8), (8, 8), (8, 7), (6, 6), (4, 5), (4, 4).

4. (4, 6), (4, 5), (5, 6), (3, 6), (4, 4), (6, 6), (5, 7), (4, 3), (5, 4), (5, 8), (4, 2), (6, 4).

How do the two algorithms differ from each other? (Hint: how and why do they visit cells in the maze in a different order?)

The two algorithms differ from each other in the directions in which they explored the maze. Because stack is a LIFO (Last in first out) algorithm, it will visit the maze in order of North, East, South and West directions, since our program stacks coordinate points with North being LAST upto West being stacked first. In this way, the stack is a depth first search, in that it will usually not plot the most efficient path to the maze ending, however it will search more squares. On the other hand, queues are a FIFO (First in first out) algorithm, so it searches the maze in order of West, South, East, North, since West is entered first and North is entered last on the queue. This is a characteristic of queues which use breadth search to visit the maze, meaning it may not cover as many squares as a depth search algorithm, but will always find the most efficient path to a destination.