**Q2:**

Popped: 5, 3

Popped: 6, 3

Popped: 4, 3

Popped: 4, 2

Popped: 4, 1

Popped: 3, 1

Popped: 2, 1

Popped: 1, 1

Popped: 1, 2

Popped: 3, 3

Popped: 5, 4

Popped: 5, 5

**Q4:**

Popped: 5, 3

Popped: 5, 4

Popped: 4, 3

Popped: 6, 3

Popped: 5, 5

Popped: 3, 3

Popped: 4, 2

Popped: 5, 6

Popped: 4, 5

Popped: 4, 1

Popped: 5, 7

Popped: 3, 5

The two algorithms differ from the order they search:

**The stack used depth first searching while the queue used breadth first searching.**

The stack searches this way because it is **last in, first out**.

The stack would run down a path until it got to the end and couldn't go any farther (there was either a wall or already visited before) then go back to the last place it had to make a decision in the last intersection.

The queue operated this way because it is **first in, first out**.

The queue would move one step forward in all its branches/paths, it found each turn until the path reached a dead end.