**Brian Chang**

**UID: 405-142-813**

**CS32 - Homework 2 doc**

**2. Given the algorithm, main function, and maze shown at the end of problem 1, what are the first 12 (r,c) coordinates popped off the stack by the algorithm?**

1. (3,5)
2. (3,6)
3. (3,4)
4. (2,4)
5. (1,4)
6. (1,3)
7. (1,2)
8. (1,1)
9. (2,1)
10. (3,3)
11. (4,5)
12. (5,5)

**4. Given the same main function and maze as are shown at the end of problem 1, what are the first 12 (r,c) coordinates popped from the queue in your queue-based algorithm?**

1. (3,5)
2. (4,5)
3. (3,4)
4. (3,6)
5. (5,5)
6. (3,3)
7. (2,4)
8. (6,5)
9. (5,4)
10. (1,4)
11. (7,5)
12. (5,3)

**\*Question 4 continued on next page.**

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

Stacks and Queues are inherently different ADTs because of the order by which they “add” and “remove” items. Stacks utilise the Last In First Out (LIFO) method. This means that after each while loop, we use the most recently pushed coordinate as our new starting coordinate to branch out from.

Queues, on the other hand, utilise the First In First Out (FIFO) method. This means that after each while loop, we use the oldest pushed coordinate, which is in the “front” of the queue (or “bottom” of the stack), as our new starting coordinate to branch out from.

So, since our search algorithm runs South, West, North, East. Assuming we were given a starting where all four moves we valid. The Stack algorithm would continue its next “search” from East coordinate, whereas the Queue algorithm would continue its next “search” from the South coordinate. This results in the difference between search steps (as shown by the popped coordinates).