JonDavid Ebadirad
SER222 Summer B
Unit 3 Linked Structures Writeup

Goal

The goal of this assignment was to use the code base given for the maze, and have the program figure out a path through the maze if the maze was solvable.

Testing

For this project I did a mix of unit testing, and input/output testing to verify that my code worked well. I unit tested each method in the classes I created to ensure that they performed how I was expecting them to perform. For each method there was at least 1 test method created. Some methods had multiple test methods that would touch on some edge cases, such as thrown exceptions or when critical information was missing. In the unit testing I would simulate a situation where you would want to use the method in question, then I would assert key values to ensure that my methods were running correctly. Unfortunately, I had to rely on other methods to setup the simulations correctly, which typically I do not want to do to make sure I am testing a single bit of functionality at a time. I did not test constructors, or the Position class (I just added a single extra constructor to it). There was a couple of methods inside the Maze class that I did not test either, these methods were copied methods that are already established as working, with a minor change in a single value (specifically setting or checking the value of the current Position in the grid).

My input and output testing was how I tested the new methods in the MazeSolver class. I started using "testfile.txt," to make sure I got the simplest of mazes solved before trying the larger ones. Once I realized that the larger mazes gave me a stack overflow, I decided to modify "testfile.txt" to include more potential paths. To the side is the final solution for testfile.txt.

```
30000
31200
33110
03330
00033

(0, 0)
(1, 0)
(2, 0)
(2, 1)
(3, 1)
(3, 2)
(3, 3)
(4, 3)
(4, 4)
```

The next step was to get the solution working using "testfile3.txt." Below to the left is the original solution for "testfile3.txt" that was copied out of the console and pasted into word, then highlighted all the 3s to easily see the path. Below center and right are "testfile3.txt" solution using custom coordinates for start and end with the far-right picture being the coordinate list. For brevity I did not include every test with custom coordinates.

```
30011001110001111001100110010101101
33300110011100011111000010101011010
01330011110011101000111110001101001
11133333333333112220011111110000111
10011001110003331001100110010101101
11100110011100033333000010101011010
10011001110001111003100111111111111
11100110011100013333000010101011010
01110011110011103000333310001101001
10011001110001113003300330010101101
11100110011100013333000030101011010
01110011110011101000111130001101101
11111111111111111111001113111111111
11111111111111111111003333111000011
01110011110011101000131110001101001
11111111111111111111003333333000011
10011001110001111001100110033331111
11100110011100011111000020101033010
10011001110001111001100220010103301
11100110011100011111000020101011333
01110011110011101000222220001101103
```

```
10011001110001111001100110010101101
11100110011100011111000010101011010
01330011110011101000111110001101001
11133333333333112220011111110000111
10011001110003331001100110010101101
11100110011100033333000010101011010
10011001110001111003100111111111111
11100110011100013333000010101011010
01110011110011103000111110001101001
10011001110001113003110011001010101
11100110011100013111000010101011010
01110011110011103000111110001101101
11111111111111113110011111111111111
11111111111113333110011111110000111
01110011110011101000111110001101001
11111111111111111110011111110000111
10011001110001111001100110011111111
11100110011100011111000010101011010
10011001110001111001100110010101101
11100110011100011111000010101011111
01110011110011101000111110001101101
```

```
(2, 2)
(2, 3)
(3, 3)
(3, 4)
(3, 5)
(3, 6)
(3, 7)
(3, 8)
(3, 9)
(3, 10)
(3, 11)
(3, 12)
(3, 13)
(4, 13)
(4, 14)
(4, 15)
(5, 15)
(5, 16)
(5, 17)
(5, 18)
(5, 19)
(6, 19)
(7, 19)
(7, 18)
(7, 17)
(7, 16)
(8, 16)
(9, 16)
(10, 16)
(11, 16)
(12, 16)
(13, 16)
(13, 15)
(13, 14)
(13, 13)
```

JonDavid Ebadirad
SER222 Summer B
Unit 3 Linked Structures Writeup

Lastly for this project I needed something special to try so I had confidence that this wasn't a coincidence. I copied the maze array from https://github.com/oppenheimj/maze-generator and imported it into "testfile4.txt." I figured this would be the least bias option and would ensure I had a solid maze solver. Below are the results using custom coordinates.

```
Enter the name of the file containing the maze: testfile4.txt
Would you like to define a starting position and an ending position?(y/n)y
Enter the starting position (two integers separated by a comma).
x must be between 0 and 29 and y must be between 0 and 29: 14,3
Enter the ending position (two integers separated by a comma).
x must be between 0 and 29 and y must be between 0 and 29: 29,29
The maze was successfully traversed!
```

```
10110001111011133330133330101 1
10011111001100030030030030111 0
11010000000111030030030330101 0
01011101101101033030330300101 1
11000101001001003033300330100 0
10111101111101003000122030111 0
10100000000001103330000030001 1
11100111111111010003333303330 01
00011100000000111100003000330 1
22010101011111000101133010030 1
02010111010100001100030011030 1
22010000110101111000330010033 1
20010011100111000033301110003 0
22010110000000001130000010333 0
02030101133333011033011110300 1
21330100030003000003000100300 1
20300111033303333303011103300 1
00311001100300000333001003001 1
33301000110330110000111113333 0
30001000010033300100100000003 3
33300333011100330111103333000 3
10300303000101033000033003033 3
10303303110101003303330003030 0
10333033000101000333011033030 1
10000030111101000000001030033 3
10103330001003333033301033000 3
11103011011003003330300033303
01003000010333110000033301030 3
11103303330300000111000300030 3
10100333033301111101111333330 3
```

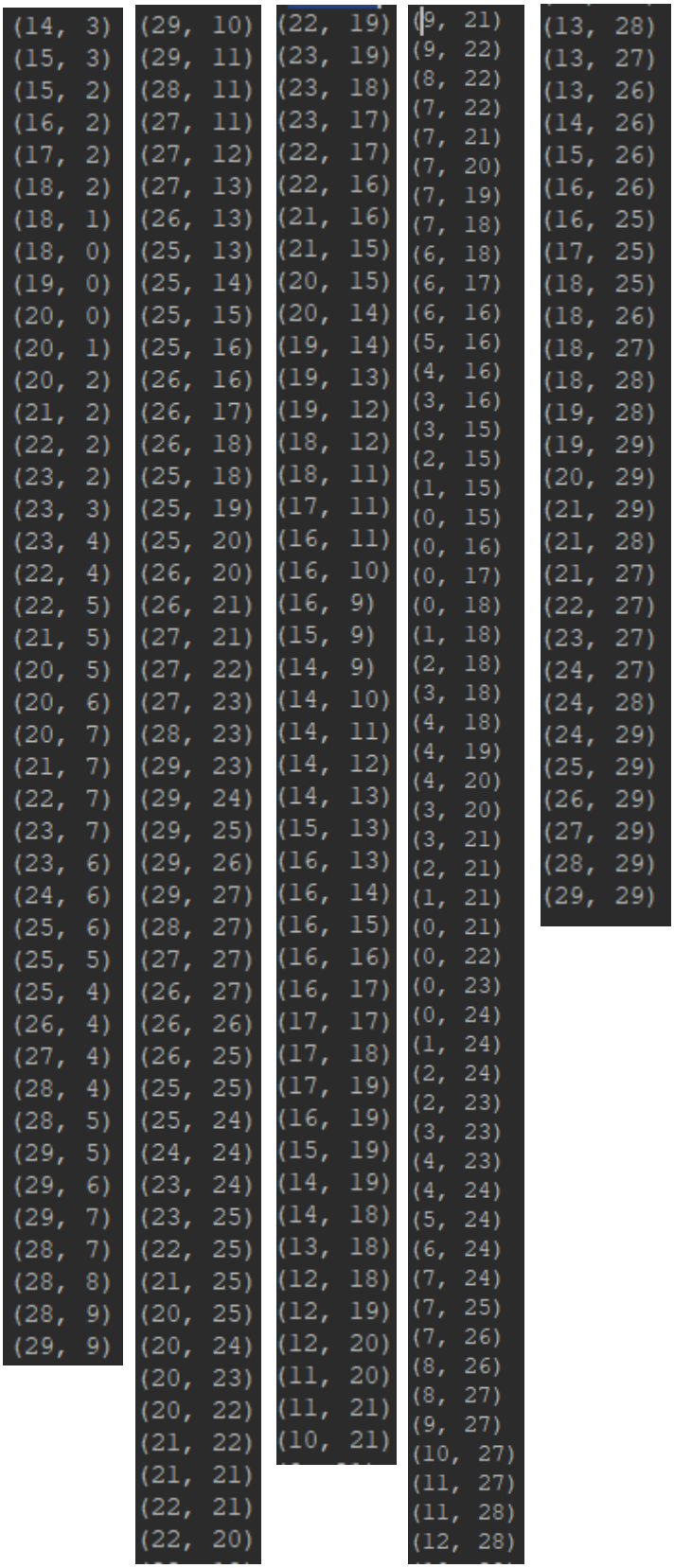
Solution in word on the left

Console output of the solution on the right.

Coordinates below

```
10110001111011133330133330101 1
10011111001100030030030030111 0
11010000000111030030030330101 0
01011101101101033030330300101 1
11000101001001003033300330100 0
10111101111101003000122030111 0
10100000000001103330000030001 1
11100111111111010003333303330 01
00011100000000111100003000330 1
22010101011111000101133010030 1
02010111010100001100030011030 1
22010000110101111000330010033 1
20010011100111000033301110003 0
22010110000000001130000010333 0
02030101133333011033011110300 1
21330100030003000003000100300 1
20300111033303333303011103300 1
00311001100300000333001003001 1
33301000110330110000111113333 0
30001000010033300100100000003 3
33300333011100330111103333000 3
10300303000101033000033003033 3
10303303110101003303330003030 0
10333033000101000333011033030 1
10000030111101000000001030033 3
10103330001003333033301033000 3
11103011011003003330300033303
01003000010333110000033301030 3
11103303330300000111000300030 3
10100333033301111101111333330 3
```

JonDavid Ebadirad
SER222 Summer B
Unit 3 Linked Structures Writeup



Coordinate output