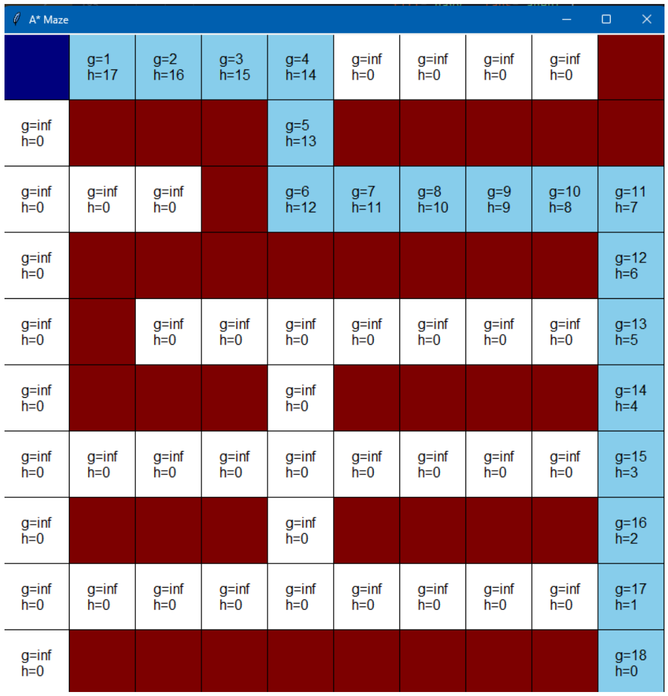
Problem 1:

Output with original A Mapping:*



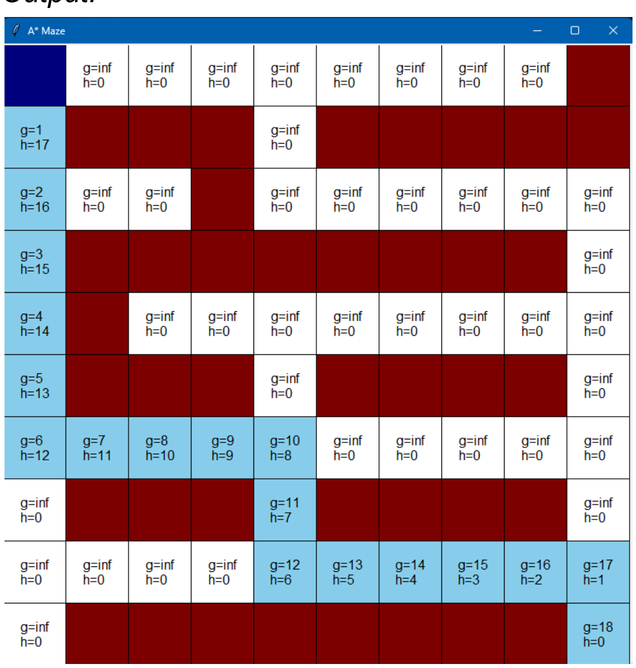
Modification to make it a Greedy Best-First Algorithm:

```
self.cells[new_pos[0]][new_pos[1]].f = new_g +
self.cells[new_pos[0]][new_pos[1]].h
```

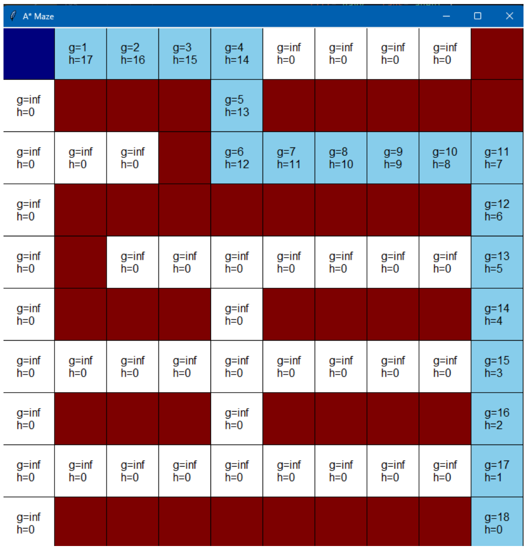
Becomes:

```
self.cells[new_pos[0]][new_pos[1]].f = self.cells[new_pos[0]][new_pos[0]].h
#Get rid of g() to make it greedy
#Changed the second [new_pos[0]] from 1 to 0 simply by messing with numbers
```

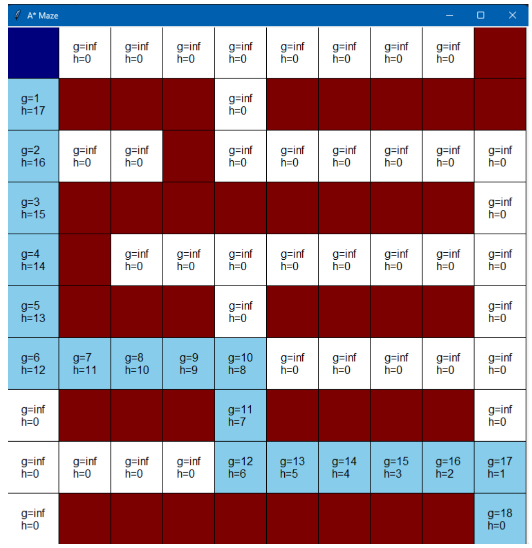
Output:



Comparison: Original vs. Greedy:



Original



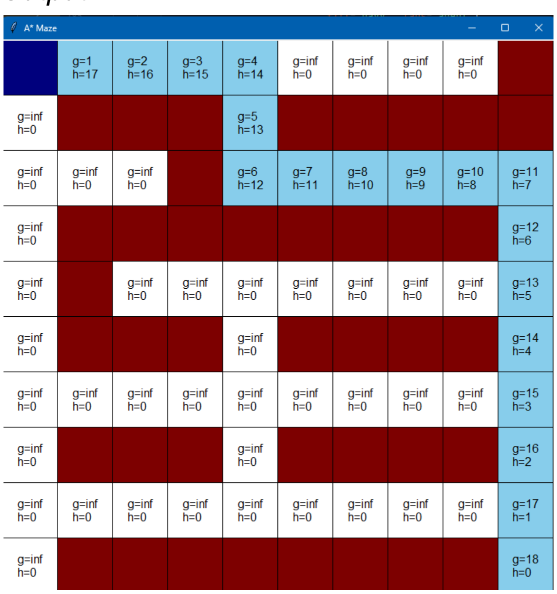
Greedy

Conclusion:
The greedy path and the A* path both take 18 moves to reach the goal. However, the A* takes the top path with a balance between the g() and h(), and the greedy path favors the h() and chooses based on the best of only that value.

Problem 2:
Original pathing makes the agent go only N, S, E, and W:

```
for dx, dy in [(0, 1), (0, -1), (1, 0), (-1, 0)]:
```

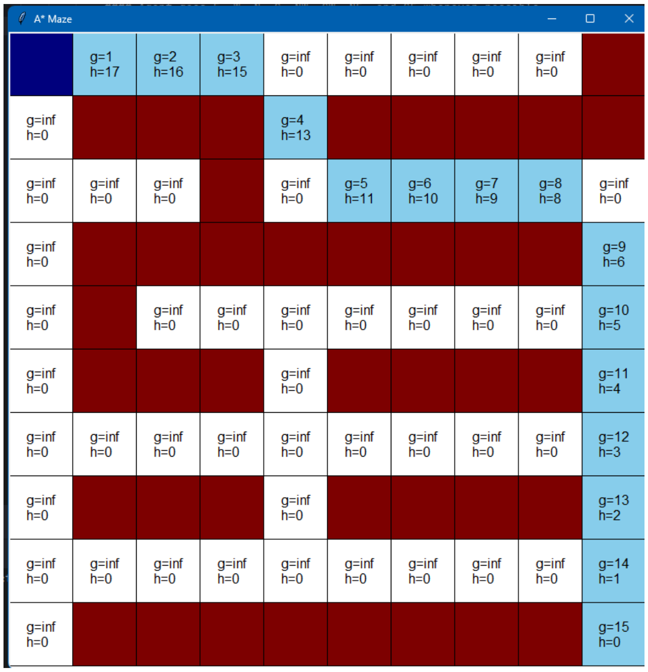
Output:



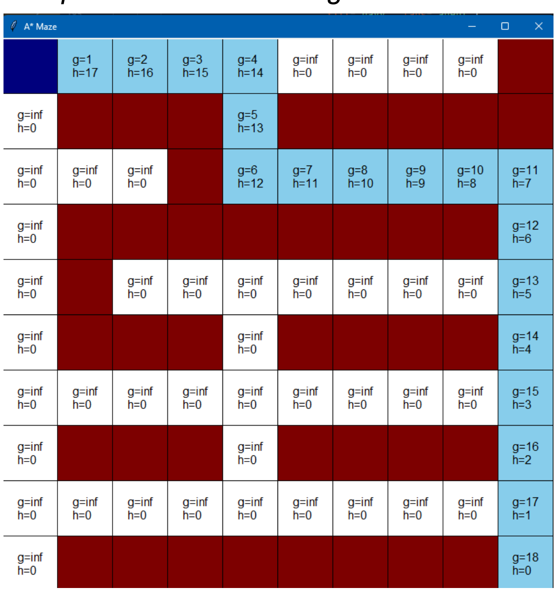
New pathing adding NE, NW, SE, and NE:

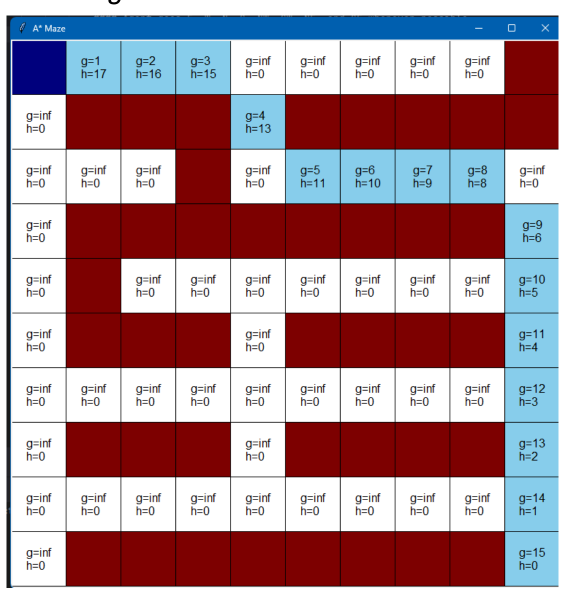
```
for dx, dy in [(0, 1), (0, -1), (1, 0), (-1, 0), (1,1), (-1, -1), (1, -1),
(-1,1)]:
```

Output:



Comparison Between Original vs. Euclidean Pathing:





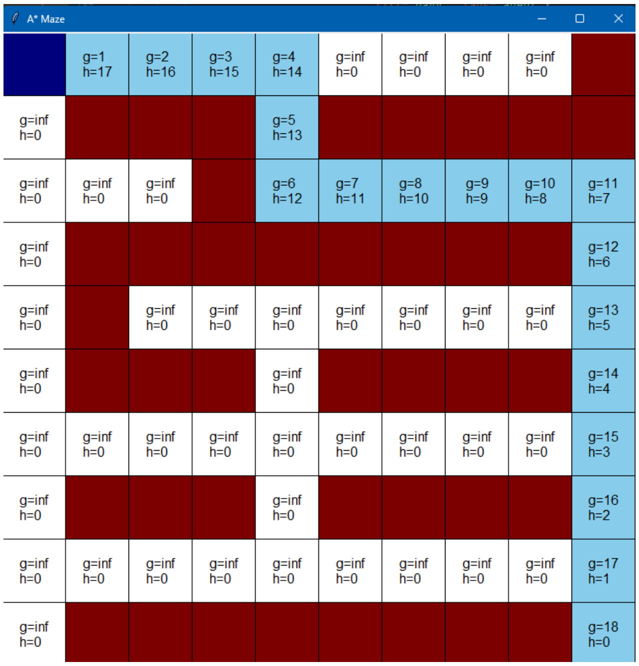
Original　　　　　　　　　　　　Euclidean

Conclusion:

The Euclidean pathing uses only 16 moves to reach the goal, while the original uses 18. This is because the Euclidean pathing can skip the corner moves the original has to take because this pathing can move diagonally.

Problem 3:

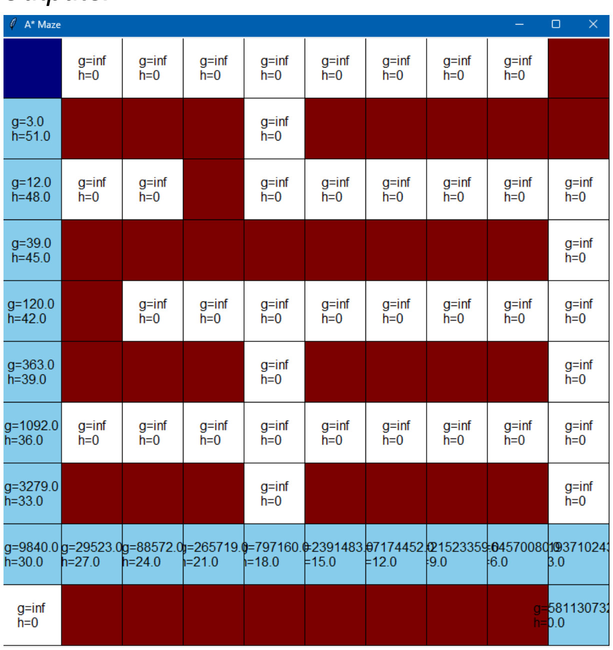
Original grid with no weights:



New code:

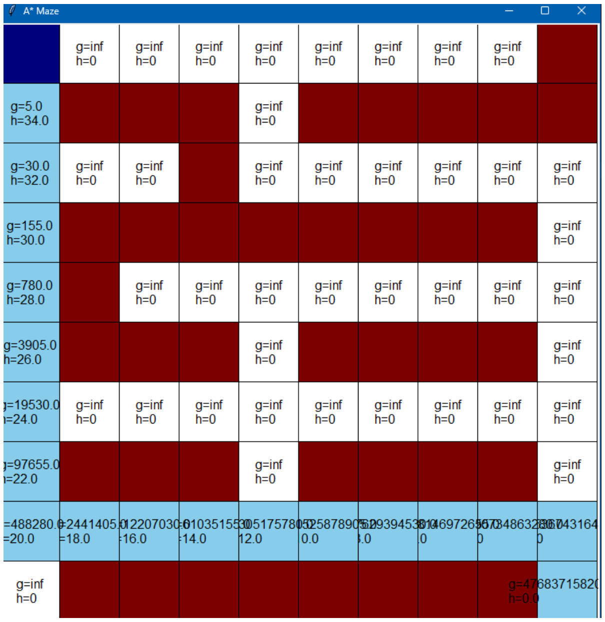
```
self.cells[new_pos[0]][new_pos[1]].g = a * new_g
self.cells[new_pos[0]][new_pos[1]].h = b * self.heuristic(new_pos)
self.cells[new_pos[0]][new_pos[1]].f = new_g +
self.cells[new_pos[0]][new_pos[1]].h
```

Outputs:

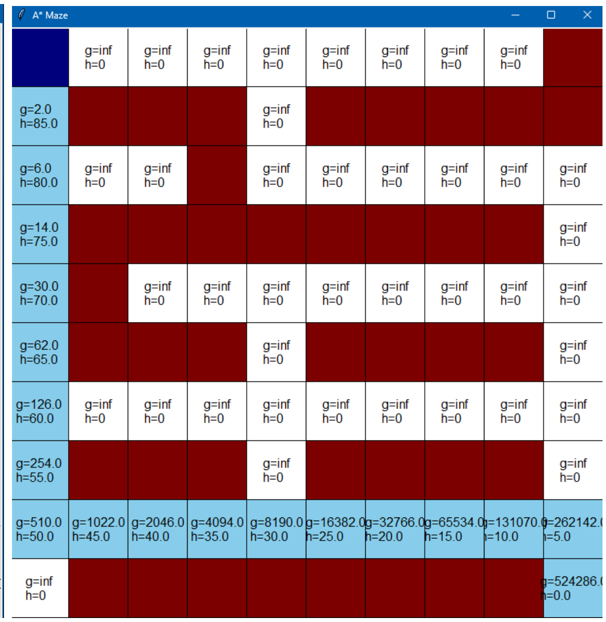


a = 3.0, b = 3.0

a = 2.0, b = 2.0



a = 5.0, b = 2.0



a = 2.0, b = 5.0

Conclusion:

The paths are all different from the original weightless grid. However, all of the weighted paths are the same, just with different numbers attached for g() and h(). This could be because the heuristic is similar in the outputs that are weighted. Possible closer to each other than to the original unweighted heuristic.