Emily Foreacre
10/20/2025
Dr. Forouraghi
AI: CSC 362
Purpose: In this assignment we explore how the A* and Greedy Best-First search algorithms compare to each other in a maze environment and how variations in heuristics, movement rules, and weighting parameters (α and β) influence the algorithms performance.

Problem 1 (10 points)

Modify **AStarMaze** to compare the behaviors of the **Greedy Best-First** and **A*** search algorithms. You need to modify the maze configuration so you can visually observe differences in the optimum paths generated by the two algorithms. Your report should include a side-by-side comparison of the two approaches similar to the graph shown below along with your explanation. You only need to draw the shortest paths and not the highlighted frontiers.

A* :

I kept the evaluation function the same.

```
### Update the evaluation function for the cell n: f(n) = g(n) + h(n)

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

I also changed the maze layout so that you could see a visible difference between A* and Greedy Best-first.

```
maze = [

    [0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0],

    [1,1,1,1,0,1,1,0,1,0,1,1,1,1,1,0,1,1,1,0],

    [0,0,0,1,0,0,0,0,1,0,0,0,0,0,1,0,0,0,1,0],

    [0,1,0,1,1,1,0,1,1,1,1,1,1,0,1,1,1,0,1,0],

    [0,1,0,0,0,1,0,0,0,0,0,0,1,0,0,0,1,0,0,0],

    [0,1,1,1,0,1,1,1,1,1,1,0,1,1,1,0,1,1,1,0],

    [0,0,0,1,0,0,0,0,0,0,1,0,0,0,1,0,0,0,1,0],

    [1,1,0,1,1,1,1,1,1,0,1,1,1,0,1,1,1,0,1,0],

    [0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0],
```

```
    [0,1,1,1,1,1,0,1,1,1,1,1,1,1,1,1,0,1,1,0],
    [0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0],
    [1,1,1,1,0,1,1,1,1,1,1,0,1,1,1,1,1,0,1,0],
    [0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0],
    [0,1,0,1,1,1,1,1,1,0,1,1,1,1,1,0,1,1,1,0],
    [0,1,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0],
    [0,1,1,1,1,1,1,0,1,1,1,1,1,0,1,1,1,1,1,0],
    [0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0],
    [1,1,1,1,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1,0],
    [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
    [0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0]
]
```

I used the AStarMaze_V1 python code as the foundation to HW3. For the first part of problem 1, the only change I made was to the maze. I figured out that if I hadn’t changed the maze then you wouldn’t be able to see the difference between A* and Greedy Best-First. I created the maze randomly, and figured out that a 20x20 was a good maze size to see the difference. You aren’t able to see the whole maze in the screenshots, but with this portion you can see the differences.

g=0 h=38	g=1 h=37	g=2 h=36	g=3 h=35	g=4 h=34	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0
				g=5 h=33			g=inf h=0		g=inf h=0						g=inf h=0				g=inf h=0
g=inf h=0	g=inf h=0	g=inf h=0		g=6 h=32	g=7 h=31	g=8 h=30	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0
g=inf h=0		g=inf h=0				g=9 h=29							g=inf h=0				g=inf h=0		g=inf h=0
g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0		g=10 h=28	g=11 h=27	g=12 h=26	g=13 h=25	g=14 h=24	g=15 h=23		g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0
g=inf h=0				g=inf h=0							g=16 h=22				g=inf h=0				g=inf h=0
g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=17 h=21	g=18 h=20	g=19 h=19		g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0
		g=inf h=0							g=inf h=0				g=20 h=18				g=inf h=0		g=inf h=0
g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=21 h=17	g=22 h=16	g=23 h=15	g=24 h=14	g=25 h=13	g=26 h=12	g=27 h=11
g=inf h=0						g=inf h=0										g=inf h=0			g=28 h=10

Greedy Best-First:

I changed the evaluation function.

Update the evaluation function for the cell n: f(n) = h(n)

self.cells[new_pos[0]][new_pos[1]].f = self.cells[new_pos[0]][new_pos[1]].h

g=0 h=38	g=1 h=37	g=2 h=36	g=3 h=35	g=4 h=34	g=5 h=33	g=6 h=32	g=7 h=31		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0
				g=inf h=0			g=8 h=30		g=inf h=0						g=inf h=0				g=inf h=0
g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=10 h=30	g=9 h=29		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0
g=inf h=0		g=inf h=0				g=11 h=29							g=inf h=0				g=inf h=0		g=inf h=0
g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0		g=12 h=28	g=13 h=27	g=14 h=26	g=15 h=25	g=16 h=24	g=17 h=23		g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0
g=inf h=0				g=inf h=0							g=18 h=22				g=inf h=0				g=inf h=0
g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=19 h=21	g=20 h=20	g=21 h=19		g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0
		g=inf h=0							g=inf h=0				g=22 h=18				g=inf h=0		g=inf h=0
g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=23 h=17	g=24 h=16	g=25 h=15	g=26 h=14	g=27 h=13	g=28 h=12	g=29 h=11
g=inf h=0						g=inf h=0										g=inf h=0			g=30 h=10

Greedy Best-First prioritizes nodes closer to the goal, potentially ignoring path cost, while A* balances path cost and the heuristic, often producing a more optimal path to the goal node.

Problem 2 (10 points)

Repeat the above experiment but this time:

- Use the Euclidean Distance heuristic.
- The agent is allowed to make diagonal moves (i.e., NE, NW, SE, SW) in addition to the usual N, S, E, and W moves.
- The moves are made randomly and not in any specific order.

I changed the heuristic to euclidean distance.

```
def heuristic(self, pos):
    dx = pos[0] - self.goal_pos[0]
    dy = pos[1] - self.goal_pos[1]
    return math.sqrt(dx*dx + dy*dy)
```

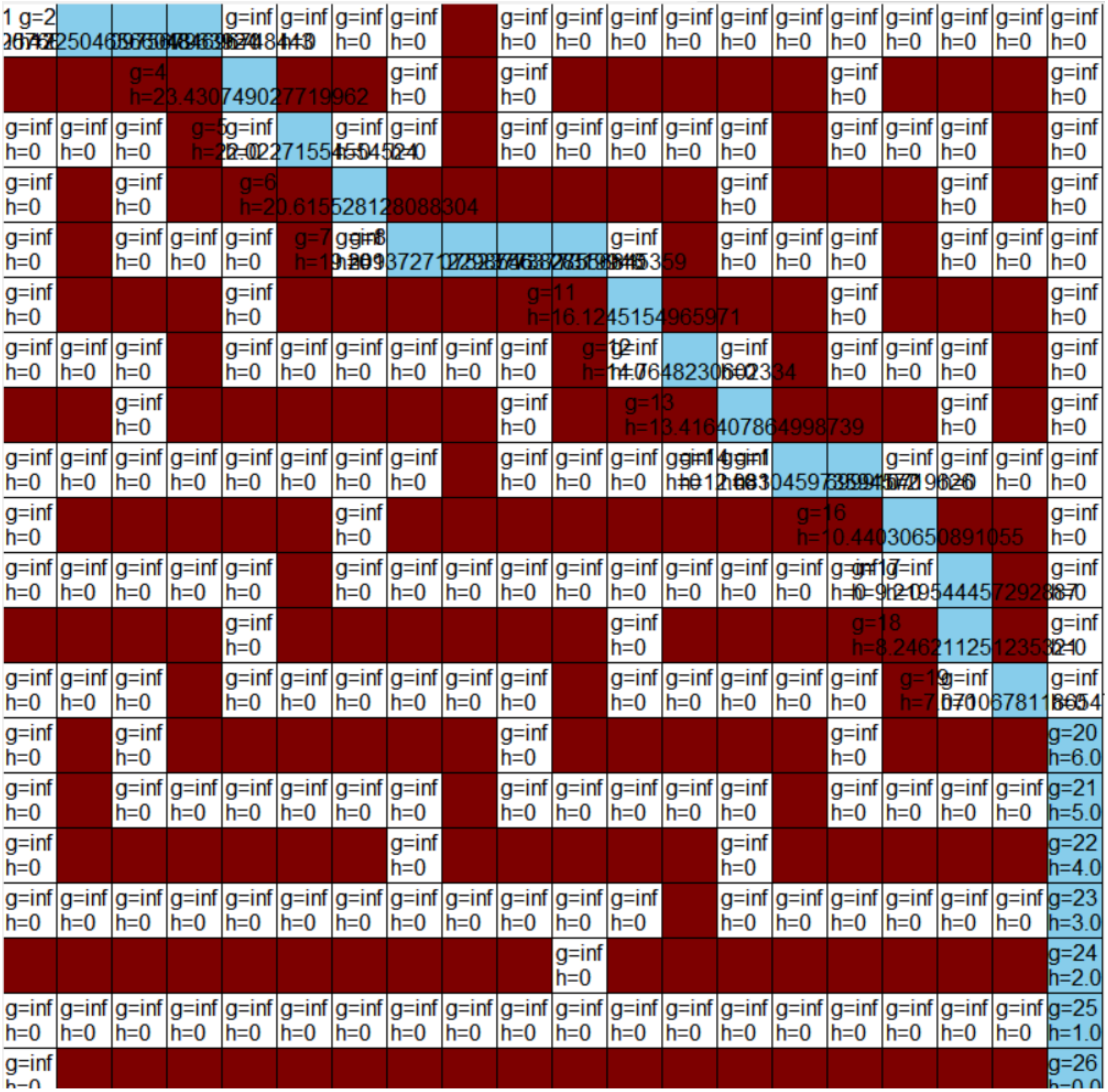
Allowed diagonal moves by adding in this bit of code.

```
neighbors = [(0,1),(0,-1),(1,0),(-1,0),  # N, S, E, W
             (1,1), (1,-1), (-1,1), (-1,-1)]  # SE, SW, NE, NW
```

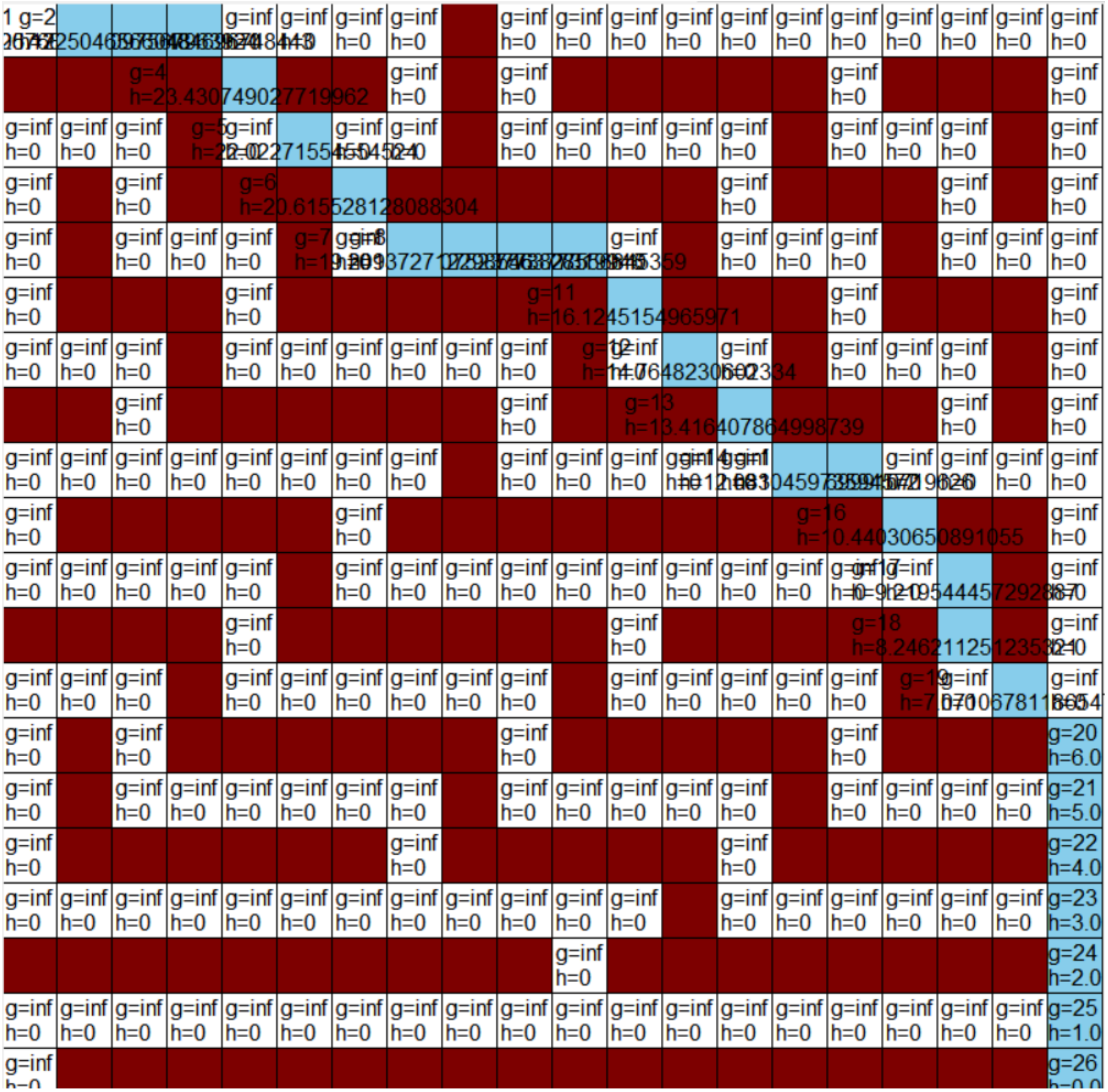
To randomize move order I added this to my code

```
random.shuffle(neighbors)
for dx, dy in neighbors:
    new_pos = (current_pos[0]+dx, current_pos[1]+dy)
```

For the **Greedy Best-First** the graph looked like this…



For the A* it looked like this…



I also changed the pixel size from self.cell_size = 75 to self.cell_size = 40, which made it so that you can see the entirety of the maze. Unfortunately, you can't see the difference between Greedy Best-First and A* when the heuristic is changed to euclidean distance. However, you can see that allowing diagonal moves and using Euclidean distance changes the path selection, and produces a shorter or more direct path compared to the Manhattan-based moves in the first problem. This cuts down the path cost from g=38 and g=40 to g=26.

Problem 3 (10 points)

The evaluation function in **AstarMaze** is defined as **f(n) = g(n) + h(n)**. A weighted version of the function can be defined as:

f(n) = α . g(n) + β. h(n) where α,β≥0

For this problem I changed/added a few new things to the foundation AStarMaze code.

```
class MazeGame:
    def __init__(self, root, maze, alpha=1, beta=1):
        self.root = root
        self.maze = maze
        self.alpha = alpha
        self.beta = beta
```

I also added to the start states initial values, self.cells[self.agent_pos[0]][self.agent_pos[1]].f = self.alpha * 0 + self.beta * self.heuristic(self.agent_pos).

Finally, I updated the evaluation function, self.cells[new_pos[0]][new_pos[1]].f = self.alpha * new_g + self.beta * self.cells[new_pos[0]][new_pos[1]].h, by adding the alpha and beta values.

α represents the weight of the path cost, and β represents the weight of the heuristic.

1. Explain how different values of α and β affect the A* algorithm's behavior. Tabulate your results:

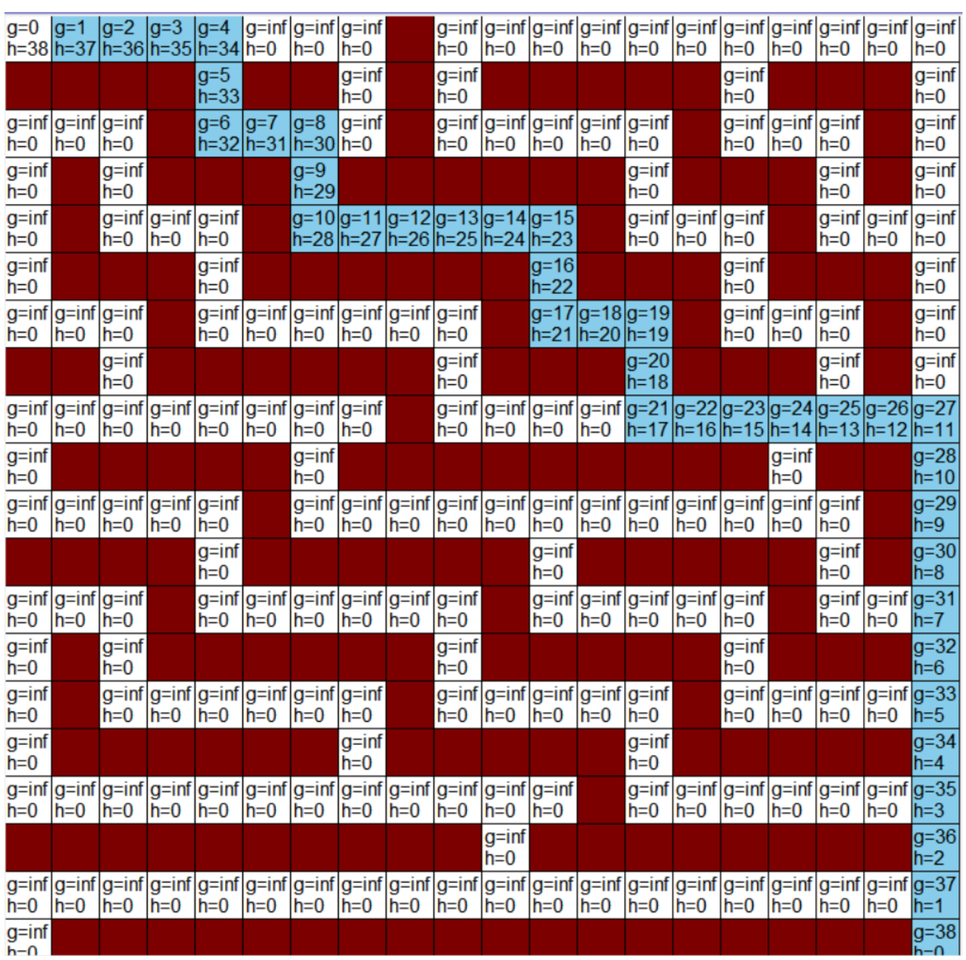
α	β	**Observed Behavior/Explanation**
1	**1**	**Seems to have found the optimal path with a g=38 when h=0. This looks to be a A* algorithm; with the heuristic and cost being balanced.**

1	**0**	**Also has a g=38 when h=0. Because it only follows the path cost, this acts as a UCS algorithm.**
0	**1**	**Didn’t find the optimal path. Seems to be a Greedy Best-First approach; with g=40 when h=0. Only follows the heuristic, and ignores cost.**
1	**2**	**Was the same as the Greedy Best-First approach. Favors the heuristic over the cost; with g=40 when h=0.**

2. β can be considered the algorithm's bias towards states that are closer to goal. Run the algorithm for various values of the bias to determine what changes, if any, are observed in the optimum path. Include screenshots of the path for each specific value of β along with your explanation.

Here I have added my screenshots of the graphs for the values of β. Some graphs were the same so I made sure to clarify which ones were the same at the top of the graph. I added my explanations to the Observation table.

(α= 1, β=1) and (α= 1, β=0)



g=0 h=38	g=1 h=37	g=2 h=36	g=3 h=35	g=4 h=34	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0
				g=5 h=33			g=inf h=0		g=inf h=0						g=inf h=0				g=inf h=0
g=inf h=0	g=inf h=0	g=inf h=0		g=6 h=32	g=7 h=31	g=8 h=30	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0
g=inf h=0		g=inf h=0				g=9 h=29							g=inf h=0				g=inf h=0		g=inf h=0
g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0		g=10 h=28	g=11 h=27	g=12 h=26	g=13 h=25	g=14 h=24	g=15 h=23		g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0
g=inf h=0				g=inf h=0							g=16 h=22				g=inf h=0				g=inf h=0
g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=17 h=21	g=18 h=20	g=19 h=19		g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0
		g=inf h=0							g=inf h=0				g=20 h=18				g=inf h=0		g=inf h=0
g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=21 h=17	g=22 h=16	g=23 h=15	g=24 h=14	g=25 h=13	g=26 h=12	g=27 h=11
g=inf h=0						g=inf h=0										g=inf h=0			g=28 h=10
g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=29 h=9
				g=inf h=0							g=inf h=0						g=inf h=0		g=30 h=8
g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=31 h=7
g=inf h=0		g=inf h=0							g=inf h=0						g=inf h=0				g=32 h=6
g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=33 h=5
g=inf h=0							g=inf h=0						g=inf h=0						g=34 h=4
g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=35 h=3
										g=inf h=0									g=36 h=2
g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=37 h=1
g=inf h=0																			g=38 h=0

(α= 0, β=1) and (α= 1, β=2)

g=0 h=38	g=1 h=37	g=2 h=36	g=3 h=35	g=4 h=34	g=5 h=33	g=6 h=32	g=7 h=31		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0
				g=inf h=0			g=8 h=30		g=inf h=0						g=inf h=0				g=inf h=0
g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=10 h=30	g=9 h=29		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0
g=inf h=0		g=inf h=0				g=11 h=29							g=inf h=0				g=inf h=0		g=inf h=0
g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0		g=12 h=28	g=13 h=27	g=14 h=26	g=15 h=25	g=16 h=24	g=17 h=23		g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0
g=inf h=0				g=inf h=0							g=18 h=22				g=inf h=0				g=inf h=0
g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=19 h=21	g=20 h=20	g=21 h=19		g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0
		g=inf h=0							g=inf h=0				g=22 h=18				g=inf h=0		g=inf h=0
g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=23 h=17	g=24 h=16	g=25 h=15	g=26 h=14	g=27 h=13	g=28 h=12	g=29 h=11
g=inf h=0						g=inf h=0										g=inf h=0			g=30 h=10
g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=31 h=9
				g=inf h=0							g=inf h=0						g=inf h=0		g=32 h=8
g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=33 h=7
g=inf h=0		g=inf h=0							g=inf h=0						g=inf h=0				g=34 h=6
g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=35 h=5
g=inf h=0							g=inf h=0						g=inf h=0						g=36 h=4
g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0		g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=37 h=3
										g=inf h=0									g=38 h=2
g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=inf h=0	g=39 h=1
g=inf h=0																			g=40 h=0

Conclusion:
Across all problems 1-3 in the homework, the A* search algorithm demonstrated that it can find the optimal solution for the maze because it balanced the g(n), or actual path cost, to the h(n), heuristic. Overall, the homework problems showed that A*'s performance depends on what heuristic function you use, and balancing the actual path cost and the heuristic (bias). When you use these parameters correctly, A* can be optimal and efficient.