```
Results on running python autograder.py

Question q1: Finding a Fixed Food Dot using Depth First
Search
==========

*** PASS: test_cases/q1/graph_backtrack.test
***    solution:     ['1:A->C', '0:C->G']
***    expanded_states: ['A', 'D', 'C']
*** PASS: test_cases/q1/graph_bfs_vs_dfs.test
***    solution:     ['2:A->D', '0:D->G']
***    expanded_states: ['A', 'D']
*** PASS: test_cases/q1/graph_infinite.test
***    solution:     ['0:A->B', '1:B->C', '1:C->G']
***    expanded_states: ['A', 'B', 'C']
*** PASS: test_cases/q1/graph_manypaths.test
***    solution:     ['2:A->B2', '0:B2->C', '0:C->D',
'2:D->E2', '0:E2->F', '0:F->G']
***    expanded_states: ['A', 'B2', 'C', 'D', 'E2', 'F']
*** PASS: test_cases/q1/pacman_1.test
***    pacman layout:      mediumMaze
***    solution length: 130
***    nodes expanded:     146

### Question q1: 3/3 ###


Question q2: Breadth First Search
==========

*** PASS: test_cases/q2/graph_backtrack.test
***    solution:     ['1:A->C', '0:C->G']
***    expanded_states: ['A', 'B', 'C', 'D']
*** PASS: test_cases/q2/graph_bfs_vs_dfs.test
***    solution:     ['1:A->G']
***    expanded_states: ['A', 'B']
*** PASS: test_cases/q2/graph_infinite.test
***    solution:     ['0:A->B', '1:B->C', '1:C->G']
***    expanded_states: ['A', 'B', 'C']
*** PASS: test_cases/q2/graph_manypaths.test
```

```
***    solution:    ['1:A->C', '0:C->D', '1:D->F',
'0:F->G']
***    expanded_states: ['A', 'B1', 'C', 'B2', 'D',
'E1', 'F', 'E2']
*** PASS: test_cases/q2/pacman_1.test
***    pacman layout:      mediumMaze
***    solution length: 68
***    nodes expanded:     269

### Question q2: 3/3 ###


Question q3: Varying the Cost Function
==========

*** PASS: test_cases/q3/graph_backtrack.test
***    solution:    ['1:A->C', '0:C->G']
***    expanded_states: ['A', 'B', 'C', 'D']
*** PASS: test_cases/q3/graph_bfs_vs_dfs.test
***    solution:    ['1:A->G']
***    expanded_states: ['A', 'B']
*** PASS: test_cases/q3/graph_infinite.test
***    solution:    ['0:A->B', '1:B->C', '1:C->G']
***    expanded_states: ['A', 'B', 'C']
*** PASS: test_cases/q3/graph_manypaths.test
***    solution:    ['1:A->C', '0:C->D', '1:D->F',
'0:F->G']
***    expanded_states: ['A', 'B1', 'C', 'B2', 'D',
'E1', 'F', 'E2']
*** PASS: test_cases/q3/ucs_0_graph.test
***    solution:    ['Right', 'Down', 'Down']
***    expanded_states: ['A', 'B', 'D', 'C', 'G']
*** PASS: test_cases/q3/ucs_1_problemC.test
***    pacman layout:      mediumMaze
***    solution length: 68
***    nodes expanded:     269
*** PASS: test_cases/q3/ucs_2_problemE.test
***    pacman layout:      mediumMaze
***    solution length: 74
***    nodes expanded:     260
```

```
*** PASS: test_cases/q3/ucs_3_problemW.test
***    pacman layout:      mediumMaze
***    solution length: 152
***    nodes expanded:     173
*** PASS: test_cases/q3/ucs_4_testSearch.test
***    pacman layout:      testSearch
***    solution length: 7
***    nodes expanded:     14
*** PASS: test_cases/q3/ucs_5_goalAtDequeue.test
***    solution:    ['1:A->B', '0:B->C', '0:C->G']
***    expanded_states: ['A', 'B', 'C']

### Question q3: 3/3 ###


Question q4: A* search
==========

*** PASS: test_cases/q4/astar_0.test
***    solution:    ['Right', 'Down', 'Down']
***    expanded_states: ['A', 'B', 'D', 'C', 'G']
*** PASS: test_cases/q4/astar_1_graph_heuristic.test
***    solution:    ['0', '0', '2']
***    expanded_states: ['S', 'A', 'D', 'C']
*** PASS: test_cases/q4/astar_2_manhattan.test
***    pacman layout:      mediumMaze
***    solution length: 68
***    nodes expanded:     221
*** PASS: test_cases/q4/astar_3_goalAtDequeue.test
***    solution:    ['1:A->B', '0:B->C', '0:C->G']
***    expanded_states: ['A', 'B', 'C']
*** PASS: test_cases/q4/graph_backtrack.test
***    solution:    ['1:A->C', '0:C->G']
***    expanded_states: ['A', 'B', 'C', 'D']
*** PASS: test_cases/q4/graph_manypaths.test
***    solution:    ['1:A->C', '0:C->D', '1:D->F',
'0:F->G']
***    expanded_states: ['A', 'B1', 'C', 'B2', 'D',
'E1', 'F', 'E2']
```

```
### Question q4: 3/3 ###


Question q5: Finding All the Corners
===========

*** PASS: test_cases/q5/corner_tiny_corner.test
***    pacman layout:      tinyCorner
***    solution length:    28

### Question q5: 3/3 ###


Question q6: Corners Problem: Heuristic
===========

*** PASS: heuristic value less than true cost at start
state
*** PASS: heuristic value less than true cost at start
state
*** PASS: heuristic value less than true cost at start
state
path: ['North', 'East', 'East', 'East', 'East',
'North', 'North', 'West', 'West', 'West', 'West',
'West', 'West', 'South', 'South', 'South', 'West',
'West', 'North', 'East', 'East', 'North', 'North',
'North', 'North', 'East', 'East', 'North', 'North',
'North', 'North', 'North', 'North', 'West', 'West',
'West', 'West', 'South', 'South', 'East', 'East',
'East', 'East', 'South', 'South', 'South', 'South',
'South', 'South', 'East', 'East', 'East', 'East',
'East', 'East', 'South', 'South', 'East', 'East',
'East', 'East', 'East', 'North', 'North', 'East',
'East', 'North', 'North', 'East', 'East', 'North',
'North', 'East', 'East', 'East', 'East', 'South',
'South', 'South', 'South', 'East', 'East', 'North',
'North', 'East', 'East', 'South', 'South', 'South',
'South', 'South', 'North', 'North', 'North', 'North',
'North', 'North', 'North', 'West', 'West', 'North',
'North', 'East', 'East', 'North', 'North']
```

```
path length: 106
*** PASS: Heuristic resulted in expansion of 1136 nodes

### Question q6: 3/3 ###


Question q7: Eating All The Dots
================================

*** PASS: test_cases/q7/food_heuristic_1.test
*** PASS: test_cases/q7/food_heuristic_10.test
*** PASS: test_cases/q7/food_heuristic_11.test
*** PASS: test_cases/q7/food_heuristic_12.test
*** PASS: test_cases/q7/food_heuristic_13.test
*** PASS: test_cases/q7/food_heuristic_14.test
*** PASS: test_cases/q7/food_heuristic_15.test
*** PASS: test_cases/q7/food_heuristic_16.test
*** PASS: test_cases/q7/food_heuristic_17.test
*** PASS: test_cases/q7/food_heuristic_2.test
*** PASS: test_cases/q7/food_heuristic_3.test
*** PASS: test_cases/q7/food_heuristic_4.test
*** PASS: test_cases/q7/food_heuristic_5.test
*** PASS: test_cases/q7/food_heuristic_6.test
*** PASS: test_cases/q7/food_heuristic_7.test
*** PASS: test_cases/q7/food_heuristic_8.test
*** PASS: test_cases/q7/food_heuristic_9.test
*** PASS: test_cases/q7/
food_heuristic_grade_tricky.test
***    expanded nodes: 4137
***    thresholds: [15000, 12000, 9000, 7000]

### Question q7: 5/4 ###


Question q8: Suboptimal Search
==============================

[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
*** PASS: test_cases/q8/closest_dot_1.test
```

```
***    pacman layout:      Test 1
***    solution length:    1
[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
*** PASS: test_cases/q8/closest_dot_10.test
***    pacman layout:      Test 10
***    solution length:    1
[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
*** PASS: test_cases/q8/closest_dot_11.test
***    pacman layout:      Test 11
***    solution length:    2
[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
*** PASS: test_cases/q8/closest_dot_12.test
***    pacman layout:      Test 12
***    solution length:    3
[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
*** PASS: test_cases/q8/closest_dot_13.test
***    pacman layout:      Test 13
***    solution length:    1
[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
*** PASS: test_cases/q8/closest_dot_2.test
***    pacman layout:      Test 2
***    solution length:    1
[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
*** PASS: test_cases/q8/closest_dot_3.test
***    pacman layout:      Test 3
***    solution length:    1
[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
*** PASS: test_cases/q8/closest_dot_4.test
***    pacman layout:      Test 4
***    solution length:    3
[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
*** PASS: test_cases/q8/closest_dot_5.test
```

```
***    pacman layout:      Test 5
***    solution length:    1
[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
*** PASS: test_cases/q8/closest_dot_6.test
***    pacman layout:      Test 6
***    solution length:    2
[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
*** PASS: test_cases/q8/closest_dot_7.test
***    pacman layout:      Test 7
***    solution length:    1
[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
*** PASS: test_cases/q8/closest_dot_8.test
***    pacman layout:      Test 8
***    solution length:    1
[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
*** PASS: test_cases/q8/closest_dot_9.test
***    pacman layout:      Test 9
***    solution length:    1

### Question q8: 3/3 ###


Finished at 13:53:05

Provisional grades
==================
Question q1: 3/3
Question q2: 3/3
Question q3: 3/3
Question q4: 3/3
Question q5: 3/3
Question q6: 3/3
Question q7: 5/4
Question q8: 3/3
------------------
Total: 26/25
```