CS440 Assignment 1 Search Report

Ruoxi Yang

Credit hour: 3

Part 1.1 Implementation Overview

State Transition:
First of all, I load the maze into a double lists, find the start and end of path. Then create 4 algorithms (DFS, BFS, Greedy Best-First, A*), which each takes in maze and start and end, and return the path and number of nodes it expands.

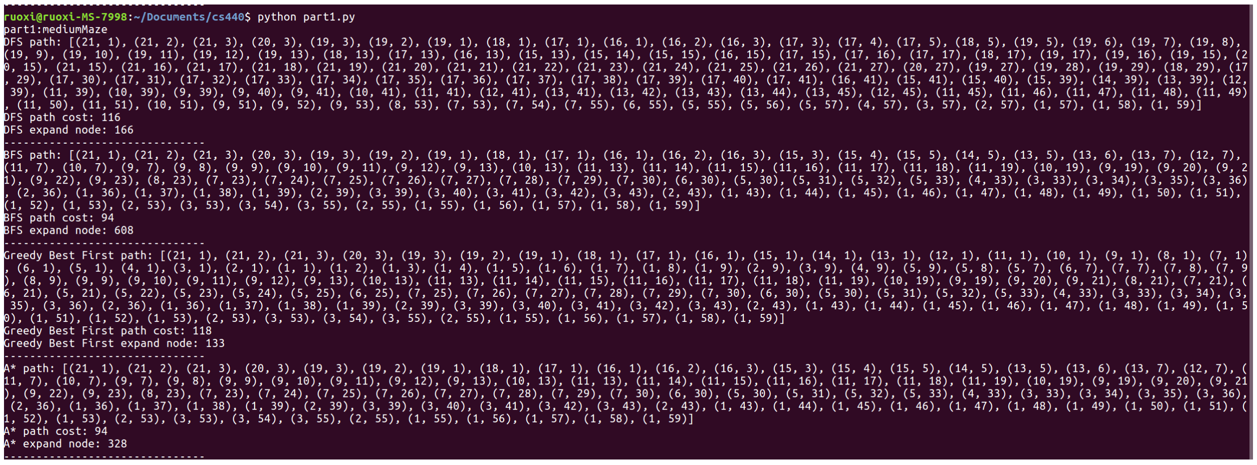
Algorithms:
DFS: Implement using stack, starting from start point, pop current node from stack (FILO), keep exploring the neighbor of current node and push them into stack until we find the end point.

BFS: Implement using queue, similar to DFS, starting from start point, pop current node from stack (FIFO), keep exploring the neighbor of current node and push them into queue until we find the end point.

Greedy Best-First: Implement using priority queue. For state representation, besides storing position for each node, I also store store the Manhattan Distance from current position to end point in each node as its priority, which can be used in comparison in building priority queue. Thus, we each time pop node with smallest distance to end point from priority queue, pushing the neighbor of current node into priority queue until we find the end point.

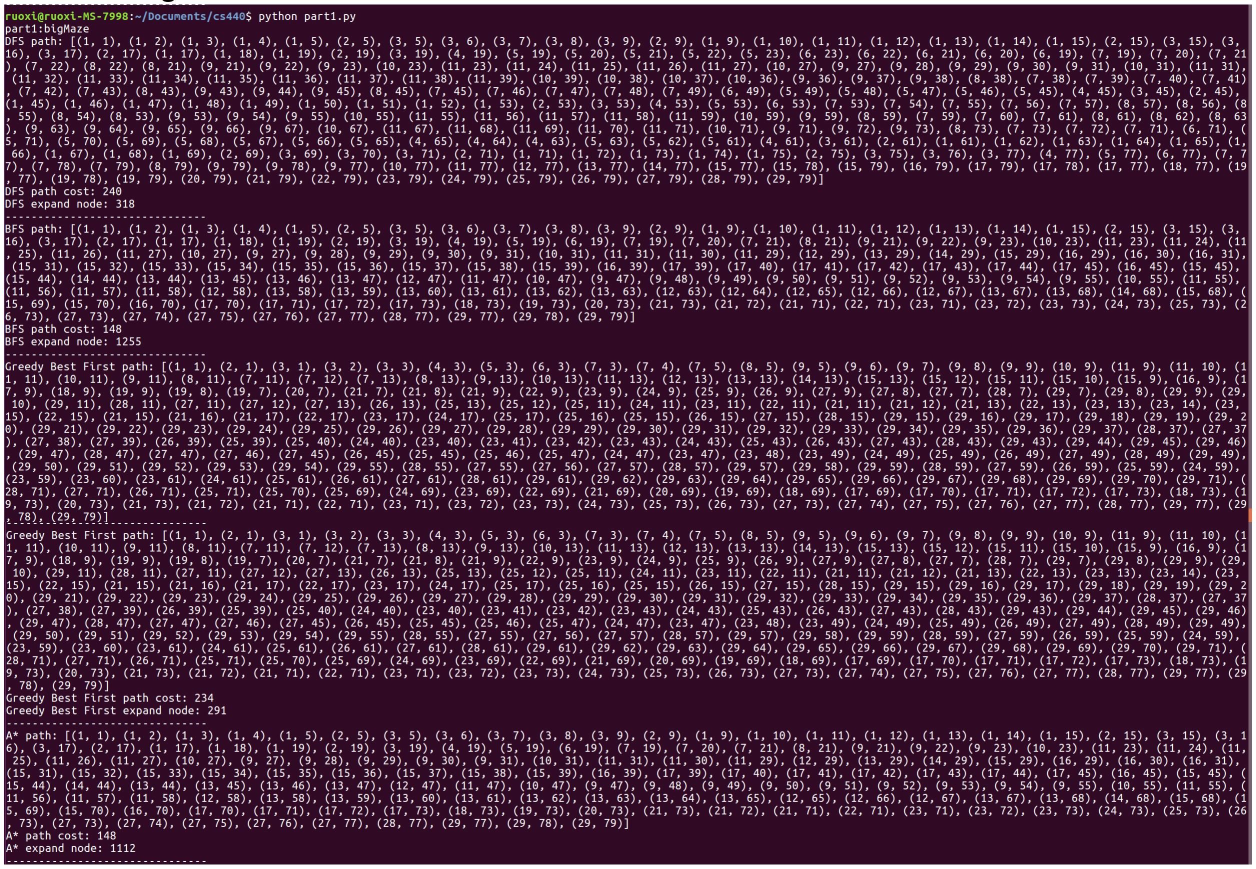
A*: Similar to Greedy Best-First. Instead of using just Manhattan Distance from current position to end point as heuristic function, I plus the cost of current path into priority of current node. This can keep track of the distance already traveled in addition to the estimated distance remaining. The other part of this algorithm is similar to Greedy Best-First.

Result for mediumMaze:



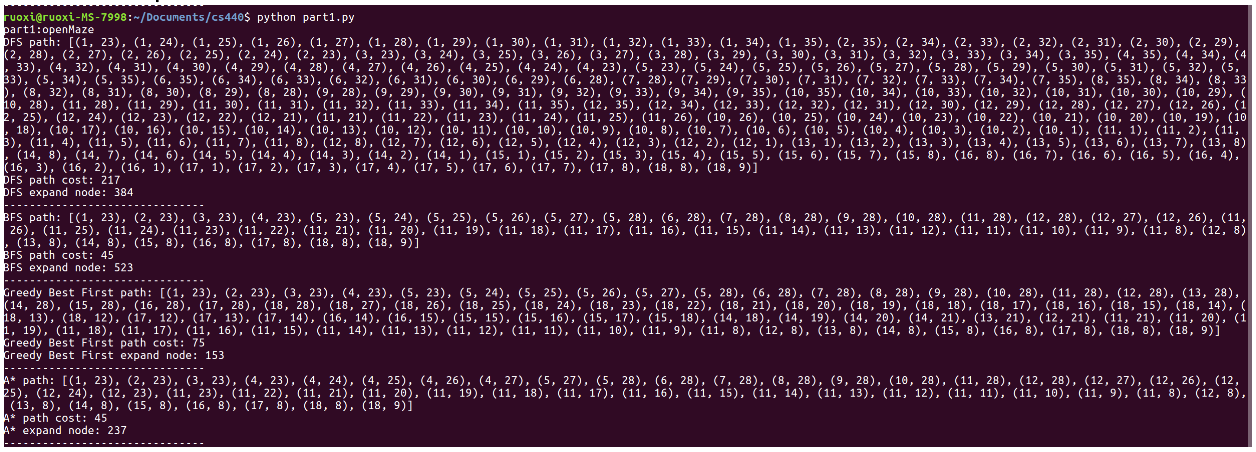
```
ruoxi@ruoxi-MS-7998:~/Documents/cs440$ python part1.py
part1:mediumMaze
DFS path: [(21, 1), (21, 2), (21, 3), (20, 3), (19, 3), (19, 2), (19, 1), (18, 1), (17, 1), (16, 1), (16, 2), (16, 3), (17, 3), (17, 4), (17, 5), (18, 5), (19, 5), (19, 6), (19, 7), (19, 8),
(19, 9), (19, 10), (19, 11), (19, 12), (19, 13), (18, 13), (17, 13), (16, 13), (15, 13), (15, 14), (15, 15), (16, 15), (17, 15), (17, 16), (17, 17), (18, 17), (19, 17), (19, 16), (19, 15), (2
0, 15), (21, 15), (21, 16), (21, 17), (21, 18), (21, 19), (21, 20), (21, 21), (21, 22), (21, 23), (21, 24), (21, 25), (21, 26), (21, 27), (20, 27), (19, 27), (19, 28), (19, 29), (18, 29), (17
, 29), (17, 30), (17, 31), (17, 32), (17, 33), (17, 34), (17, 35), (17, 36), (17, 37), (17, 38), (17, 39), (17, 40), (17, 41), (16, 41), (15, 41), (15, 40), (15, 39), (14, 39), (13, 39), (12,
 39), (11, 39), (10, 39), (9, 39), (9, 40), (9, 41), (10, 41), (11, 41), (12, 41), (13, 41), (13, 42), (13, 43), (13, 44), (13, 45), (12, 45), (11, 45), (11, 46), (11, 47), (11, 48), (11, 49)
, (11, 50), (11, 51), (10, 51), (9, 51), (9, 52), (9, 53), (8, 53), (7, 53), (7, 54), (7, 55), (6, 55), (5, 55), (5, 56), (5, 57), (4, 57), (3, 57), (2, 57), (1, 57), (1, 58), (1, 59)]
DFS path cost: 116
DFS expand node: 166
-------------------------------
BFS path: [(21, 1), (21, 2), (21, 3), (20, 3), (19, 3), (19, 2), (19, 1), (18, 1), (17, 1), (16, 1), (16, 2), (16, 3), (15, 3), (15, 4), (15, 5), (14, 5), (13, 5), (13, 6), (13, 7), (12, 7),
(11, 7), (10, 7), (9, 7), (9, 8), (9, 9), (9, 10), (9, 11), (9, 12), (9, 13), (10, 13), (11, 13), (11, 14), (11, 15), (11, 16), (11, 17), (11, 18), (11, 19), (10, 19), (9, 19), (9, 20), (9, 2
1), (9, 22), (9, 23), (8, 23), (7, 23), (7, 24), (7, 25), (7, 26), (7, 27), (7, 28), (7, 29), (7, 30), (6, 30), (5, 30), (5, 31), (5, 32), (5, 33), (4, 33), (3, 33), (3, 34), (3, 35), (3, 36)
, (2, 36), (1, 36), (1, 37), (1, 38), (1, 39), (2, 39), (3, 39), (3, 40), (3, 41), (3, 42), (3, 43), (2, 43), (1, 43), (1, 44), (1, 45), (1, 46), (1, 47), (1, 48), (1, 49), (1, 50), (1, 51),
(1, 52), (1, 53), (2, 53), (3, 53), (3, 54), (3, 55), (2, 55), (1, 55), (1, 56), (1, 57), (1, 58), (1, 59)]
BFS path cost: 94
BFS expand node: 608
-------------------------------
Greedy Best First path: [(21, 1), (21, 2), (21, 3), (20, 3), (19, 3), (19, 2), (19, 1), (18, 1), (17, 1), (16, 1), (15, 1), (14, 1), (13, 1), (12, 1), (11, 1), (10, 1), (9, 1), (8, 1), (7, 1)
, (6, 1), (5, 1), (4, 1), (3, 1), (2, 1), (1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (1, 8), (1, 9), (2, 9), (3, 9), (4, 9), (5, 9), (5, 8), (5, 7), (6, 7), (7, 7), (7, 8), (7, 9
), (8, 9), (9, 9), (9, 10), (9, 11), (9, 12), (9, 13), (10, 13), (11, 13), (11, 14), (11, 15), (11, 16), (11, 17), (11, 18), (11, 19), (10, 19), (9, 19), (9, 20), (9, 21), (8, 21), (7, 21), (
6, 21), (5, 21), (5, 22), (5, 23), (5, 24), (5, 25), (6, 25), (7, 25), (7, 26), (7, 27), (7, 28), (7, 29), (7, 30), (6, 30), (5, 30), (5, 31), (5, 32), (5, 33), (4, 33), (3, 33), (3, 34), (3,
 35), (3, 36), (2, 36), (1, 36), (1, 37), (1, 38), (1, 39), (2, 39), (3, 39), (3, 40), (3, 41), (3, 42), (3, 43), (2, 43), (1, 43), (1, 44), (1, 45), (1, 46), (1, 47), (1, 48), (1, 49), (1, 5
0), (1, 51), (1, 52), (1, 53), (2, 53), (3, 53), (3, 54), (3, 55), (2, 55), (1, 55), (1, 56), (1, 57), (1, 58), (1, 59)]
Greedy Best First path cost: 118
Greedy Best First expand node: 133
-------------------------------
A* path: [(21, 1), (21, 2), (21, 3), (20, 3), (19, 3), (19, 2), (19, 1), (18, 1), (17, 1), (16, 1), (16, 2), (16, 3), (15, 3), (15, 4), (15, 5), (14, 5), (13, 5), (13, 6), (13, 7), (12, 7), (
11, 7), (10, 7), (9, 7), (9, 8), (9, 9), (9, 10), (9, 11), (9, 12), (9, 13), (10, 13), (11, 13), (11, 14), (11, 15), (11, 16), (11, 17), (11, 18), (11, 19), (10, 19), (9, 19), (9, 20), (9, 21
), (9, 22), (9, 23), (8, 23), (7, 23), (7, 24), (7, 25), (7, 26), (7, 27), (7, 28), (7, 29), (7, 30), (6, 30), (5, 30), (5, 31), (5, 32), (5, 33), (4, 33), (3, 33), (3, 34), (3, 35), (3, 36),
 (2, 36), (1, 36), (1, 37), (1, 38), (1, 39), (2, 39), (3, 39), (3, 40), (3, 41), (3, 42), (3, 43), (2, 43), (1, 43), (1, 44), (1, 45), (1, 46), (1, 47), (1, 48), (1, 49), (1, 50), (1, 51), (
1, 52), (1, 53), (2, 53), (3, 53), (3, 54), (3, 55), (2, 55), (1, 55), (1, 56), (1, 57), (1, 58), (1, 59)]
A* path cost: 94
A* expand node: 328
-------------------------------
```

Result for bigMaze:



```
ruoxi@ruoxi-MS-7998:~/Documents/cs440$ python part1.py
part1:bigMaze
DFS path: [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 5), (3, 5), (3, 6), (3, 7), (3, 8), (3, 9), (2, 9), (1, 9), (1, 10), (1, 11), (1, 12), (1, 13), (1, 14), (1, 15), (2, 15), (3, 15), (3,
16), (3, 17), (2, 17), (1, 17), (1, 18), (1, 19), (2, 19), (3, 19), (4, 19), (5, 19), (5, 20), (5, 21), (5, 22), (5, 23), (6, 23), (6, 22), (6, 21), (6, 20), (6, 19), (7, 19), (7, 20), (7, 21
), (7, 22), (8, 22), (8, 21), (9, 21), (9, 22), (9, 23), (10, 23), (11, 23), (11, 24), (11, 25), (11, 26), (11, 27), (10, 27), (9, 27), (9, 28), (9, 29), (9, 30), (9, 31), (10, 31), (11, 31),
 (11, 32), (11, 33), (11, 34), (11, 35), (11, 36), (11, 37), (11, 38), (11, 39), (10, 39), (10, 38), (10, 37), (10, 36), (9, 36), (9, 37), (9, 38), (8, 38), (7, 38), (7, 39), (7, 40), (7, 41)
, (7, 42), (7, 43), (8, 43), (9, 43), (9, 44), (9, 45), (8, 45), (7, 45), (7, 46), (7, 47), (7, 48), (7, 49), (6, 49), (5, 49), (5, 48), (5, 47), (5, 46), (5, 45), (4, 45), (3, 45), (2, 45),
(1, 45), (1, 46), (1, 47), (1, 48), (1, 49), (1, 50), (1, 51), (1, 52), (1, 53), (2, 53), (3, 53), (4, 53), (5, 53), (6, 53), (7, 53), (7, 54), (7, 55), (7, 56), (7, 57), (8, 57), (8, 56), (8
, 55), (8, 54), (8, 53), (9, 53), (9, 54), (9, 55), (10, 55), (11, 55), (11, 56), (11, 57), (11, 58), (11, 59), (10, 59), (9, 59), (8, 59), (7, 59), (7, 60), (7, 61), (8, 61), (8, 62), (8, 63
), (9, 63), (9, 64), (9, 65), (9, 66), (9, 67), (10, 67), (11, 67), (11, 68), (11, 69), (11, 70), (11, 71), (10, 71), (9, 71), (9, 72), (9, 73), (8, 73), (7, 73), (7, 72), (7, 71), (6, 71), (
5, 71), (5, 70), (5, 69), (5, 68), (5, 67), (5, 66), (5, 65), (4, 65), (4, 64), (4, 63), (5, 63), (5, 62), (5, 61), (4, 61), (3, 61), (2, 61), (1, 61), (1, 62), (1, 63), (1, 64), (1, 65), (1,
 66), (1, 67), (1, 68), (1, 69), (2, 69), (3, 69), (3, 70), (3, 71), (2, 71), (1, 71), (1, 72), (1, 73), (1, 74), (1, 75), (2, 75), (3, 75), (3, 76), (3, 77), (4, 77), (5, 77), (6, 77), (7, 7
7), (7, 78), (7, 79), (8, 79), (9, 79), (9, 78), (9, 77), (10, 77), (11, 77), (12, 77), (13, 77), (14, 77), (15, 77), (15, 78), (15, 79), (16, 79), (17, 79), (17, 78), (17, 77), (18, 77), (19
, 77), (19, 78), (19, 79), (20, 79), (21, 79), (22, 79), (23, 79), (24, 79), (25, 79), (26, 79), (27, 79), (28, 79), (29, 79)]
DFS path cost: 240
DFS expand node: 318
-------------------------------
BFS path: [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 5), (3, 5), (3, 6), (3, 7), (3, 8), (3, 9), (2, 9), (1, 9), (1, 10), (1, 11), (1, 12), (1, 13), (1, 14), (1, 15), (2, 15), (3, 15), (3,
16), (3, 17), (2, 17), (1, 17), (1, 18), (1, 19), (2, 19), (3, 19), (4, 19), (5, 19), (6, 19), (7, 19), (7, 20), (7, 21), (8, 21), (9, 21), (9, 22), (9, 23), (10, 23), (11, 23), (11, 24), (11
, 25), (11, 26), (11, 27), (10, 27), (9, 27), (9, 28), (9, 29), (9, 30), (9, 31), (10, 31), (11, 31), (11, 30), (11, 29), (12, 29), (13, 29), (14, 29), (15, 29), (16, 29), (16, 30), (16, 31),
 (15, 31), (15, 32), (15, 33), (15, 34), (15, 35), (15, 36), (15, 37), (15, 38), (15, 39), (16, 39), (17, 39), (17, 40), (17, 41), (17, 42), (17, 43), (17, 44), (17, 45), (16, 45), (15, 45),
(15, 44), (14, 44), (13, 44), (13, 45), (13, 46), (13, 47), (12, 47), (11, 47), (10, 47), (9, 47), (9, 48), (9, 49), (9, 50), (9, 51), (9, 52), (9, 53), (9, 54), (9, 55), (10, 55), (11, 55),
(11, 56), (11, 57), (11, 58), (12, 58), (13, 58), (13, 59), (13, 60), (13, 61), (13, 62), (13, 63), (12, 63), (12, 64), (12, 65), (12, 66), (12, 67), (13, 67), (13, 68), (14, 68), (15, 68), (
15, 69), (15, 70), (16, 70), (17, 70), (17, 71), (17, 72), (17, 73), (18, 73), (19, 73), (20, 73), (21, 73), (21, 72), (21, 71), (22, 71), (23, 71), (23, 72), (23, 73), (24, 73), (25, 73), (2
6, 73), (27, 73), (27, 74), (27, 75), (27, 76), (27, 77), (28, 77), (29, 77), (29, 78), (29, 79)]
BFS path cost: 148
BFS expand node: 1255
-------------------------------
Greedy Best First path: [(1, 1), (2, 1), (3, 1), (3, 2), (3, 3), (4, 3), (5, 3), (6, 3), (7, 3), (7, 4), (7, 5), (8, 5), (9, 5), (9, 6), (9, 7), (9, 8), (9, 9), (10, 9), (11, 9), (11, 10), (1
1, 11), (10, 11), (9, 11), (8, 11), (7, 11), (7, 12), (7, 13), (8, 13), (9, 13), (10, 13), (11, 13), (12, 13), (13, 13), (14, 13), (15, 13), (15, 12), (15, 11), (15, 10), (15, 9), (16, 9), (1
7, 9), (18, 9), (19, 9), (19, 8), (19, 7), (20, 7), (21, 7), (21, 8), (21, 9), (22, 9), (23, 9), (24, 9), (25, 9), (26, 9), (27, 9), (27, 8), (27, 7), (28, 7), (29, 7), (29, 8), (29, 9), (29,
 10), (29, 11), (28, 11), (27, 11), (27, 12), (27, 13), (26, 13), (25, 13), (25, 12), (25, 11), (24, 11), (23, 11), (22, 11), (21, 11), (21, 12), (21, 13), (22, 13), (23, 13), (23, 14), (23,
15), (22, 15), (21, 15), (21, 16), (21, 17), (22, 17), (23, 17), (24, 17), (25, 17), (25, 16), (25, 15), (26, 15), (27, 15), (28, 15), (29, 15), (29, 16), (29, 17), (29, 18), (29, 19), (29, 2
0), (29, 21), (29, 22), (29, 23), (29, 24), (29, 25), (29, 26), (29, 27), (29, 28), (29, 29), (29, 30), (29, 31), (29, 32), (29, 33), (29, 34), (29, 35), (29, 36), (29, 37), (28, 37), (27, 37
), (27, 38), (27, 39), (26, 39), (25, 39), (25, 40), (24, 40), (23, 40), (23, 41), (23, 42), (23, 43), (24, 43), (25, 43), (26, 43), (27, 43), (28, 43), (29, 43), (29, 44), (29, 45), (29, 46)
, (29, 47), (28, 47), (27, 47), (27, 46), (27, 45), (26, 45), (25, 45), (25, 46), (25, 47), (24, 47), (23, 47), (23, 48), (23, 49), (24, 49), (25, 49), (26, 49), (27, 49), (28, 49), (29, 49),
 (29, 50), (29, 51), (29, 52), (29, 53), (29, 54), (29, 55), (28, 55), (27, 55), (27, 56), (27, 57), (28, 57), (29, 57), (29, 58), (29, 59), (28, 59), (27, 59), (26, 59), (25, 59), (24, 59),
(23, 59), (23, 60), (23, 61), (24, 61), (25, 61), (26, 61), (27, 61), (28, 61), (29, 61), (29, 62), (29, 63), (29, 64), (29, 65), (29, 66), (29, 67), (29, 68), (29, 69), (29, 70), (29, 71), (
28, 71), (27, 71), (26, 71), (25, 71), (25, 70), (25, 69), (24, 69), (23, 69), (22, 69), (21, 69), (20, 69), (19, 69), (18, 69), (17, 69), (17, 70), (17, 71), (17, 72), (17, 73), (18, 73), (1
9, 73), (20, 73), (21, 73), (21, 72), (21, 71), (22, 71), (23, 71), (23, 72), (23, 73), (24, 73), (25, 73), (26, 73), (27, 73), (27, 74), (27, 75), (27, 76), (27, 77), (28, 77), (29, 77), (29
, 78), (29, 79)]
-------------------------------
Greedy Best First path: [(1, 1), (2, 1), (3, 1), (3, 2), (3, 3), (4, 3), (5, 3), (6, 3), (7, 3), (7, 4), (7, 5), (8, 5), (9, 5), (9, 6), (9, 7), (9, 8), (9, 9), (10, 9), (11, 9), (11, 10), (1
1, 11), (10, 11), (9, 11), (8, 11), (7, 11), (7, 12), (7, 13), (8, 13), (9, 13), (10, 13), (11, 13), (12, 13), (13, 13), (14, 13), (15, 13), (15, 12), (15, 11), (15, 10), (15, 9), (16, 9), (1
7, 9), (18, 9), (19, 9), (19, 8), (19, 7), (20, 7), (21, 7), (21, 8), (21, 9), (22, 9), (23, 9), (24, 9), (25, 9), (26, 9), (27, 9), (27, 8), (27, 7), (28, 7), (29, 7), (29, 8), (29, 9), (29,
 10), (29, 11), (28, 11), (27, 11), (27, 12), (27, 13), (26, 13), (25, 13), (25, 12), (25, 11), (24, 11), (23, 11), (22, 11), (21, 11), (21, 12), (21, 13), (22, 13), (23, 13), (23, 14), (23,
15), (22, 15), (21, 15), (21, 16), (21, 17), (22, 17), (23, 17), (24, 17), (25, 17), (25, 16), (25, 15), (26, 15), (27, 15), (28, 15), (29, 15), (29, 16), (29, 17), (29, 18), (29, 19), (29, 2
0), (29, 21), (29, 22), (29, 23), (29, 24), (29, 25), (29, 26), (29, 27), (29, 28), (29, 29), (29, 30), (29, 31), (29, 32), (29, 33), (29, 34), (29, 35), (29, 36), (29, 37), (28, 37), (27, 37
), (27, 38), (27, 39), (26, 39), (25, 39), (25, 40), (24, 40), (23, 40), (23, 41), (23, 42), (23, 43), (24, 43), (25, 43), (26, 43), (27, 43), (28, 43), (29, 43), (29, 44), (29, 45), (29, 46)
, (29, 47), (28, 47), (27, 47), (27, 46), (27, 45), (26, 45), (25, 45), (25, 46), (25, 47), (24, 47), (23, 47), (23, 48), (23, 49), (24, 49), (25, 49), (26, 49), (27, 49), (28, 49), (29, 49),
 (29, 50), (29, 51), (29, 52), (29, 53), (29, 54), (29, 55), (28, 55), (27, 55), (27, 56), (27, 57), (28, 57), (29, 57), (29, 58), (29, 59), (28, 59), (27, 59), (26, 59), (25, 59), (24, 59),
(23, 59), (23, 60), (23, 61), (24, 61), (25, 61), (26, 61), (27, 61), (28, 61), (29, 61), (29, 62), (29, 63), (29, 64), (29, 65), (29, 66), (29, 67), (29, 68), (29, 69), (29, 70), (29, 71), (
28, 71), (27, 71), (26, 71), (25, 71), (25, 70), (25, 69), (24, 69), (23, 69), (22, 69), (21, 69), (20, 69), (19, 69), (18, 69), (17, 69), (17, 70), (17, 71), (17, 72), (17, 73), (18, 73), (1
9, 73), (20, 73), (21, 73), (21, 72), (21, 71), (22, 71), (23, 71), (23, 72), (23, 73), (24, 73), (25, 73), (26, 73), (27, 73), (27, 74), (27, 75), (27, 76), (27, 77), (28, 77), (29, 77), (29
, 78), (29, 79)]
Greedy Best First path cost: 234
Greedy Best First expand node: 291
-------------------------------
A* path: [(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (2, 5), (3, 5), (3, 6), (3, 7), (3, 8), (3, 9), (2, 9), (1, 9), (1, 10), (1, 11), (1, 12), (1, 13), (1, 14), (1, 15), (2, 15), (3, 15), (3, 1
6), (3, 17), (2, 17), (1, 17), (1, 18), (1, 19), (2, 19), (3, 19), (4, 19), (5, 19), (6, 19), (7, 19), (7, 20), (7, 21), (8, 21), (9, 21), (9, 22), (9, 23), (10, 23), (11, 23), (11, 24), (11,
 25), (11, 26), (11, 27), (10, 27), (9, 27), (9, 28), (9, 29), (9, 30), (9, 31), (10, 31), (11, 31), (11, 30), (11, 29), (12, 29), (13, 29), (14, 29), (15, 29), (16, 29), (16, 30), (16, 31),
(15, 31), (15, 32), (15, 33), (15, 34), (15, 35), (15, 36), (15, 37), (15, 38), (15, 39), (16, 39), (17, 39), (17, 40), (17, 41), (17, 42), (17, 43), (17, 44), (17, 45), (16, 45), (15, 45), (
15, 44), (14, 44), (13, 44), (13, 45), (13, 46), (13, 47), (12, 47), (11, 47), (10, 47), (9, 47), (9, 48), (9, 49), (9, 50), (9, 51), (9, 52), (9, 53), (9, 54), (9, 55), (10, 55), (11, 55), (
11, 56), (11, 57), (11, 58), (12, 58), (13, 58), (13, 59), (13, 60), (13, 61), (13, 62), (13, 63), (13, 64), (13, 65), (12, 65), (12, 66), (12, 67), (13, 67), (13, 68), (14, 68), (15, 68), (1
5, 69), (15, 70), (16, 70), (17, 70), (17, 71), (17, 72), (17, 73), (18, 73), (19, 73), (20, 73), (21, 73), (21, 72), (21, 71), (22, 71), (23, 71), (23, 72), (23, 73), (24, 73), (25, 73), (26
, 73), (27, 73), (27, 74), (27, 75), (27, 76), (27, 77), (28, 77), (29, 77), (29, 78), (29, 79)]
A* path cost: 148
A* expand node: 1112
-------------------------------
```

Result for openMaze:



```
ruoxi@ruoxi-MS-7998:~/Documents/cs440$ python part1.py
part1:openMaze
DFS path: [(1, 23), (1, 24), (1, 25), (1, 26), (1, 27), (1, 28), (1, 29), (1, 30), (1, 31), (1, 32), (1, 33), (1, 34), (1, 35), (2, 35), (2, 34), (2, 33), (2, 32), (2, 31), (2, 30), (2, 29),
(2, 28), (2, 27), (2, 26), (2, 25), (2, 24), (2, 23), (3, 23), (3, 24), (3, 25), (3, 26), (3, 27), (3, 28), (3, 29), (3, 30), (3, 31), (3, 32), (3, 33), (3, 34), (3, 35), (4, 35), (4, 34), (4
, 33), (4, 32), (4, 31), (4, 30), (4, 29), (4, 28), (4, 27), (4, 26), (4, 25), (4, 24), (4, 23), (5, 23), (5, 24), (5, 25), (5, 26), (5, 27), (5, 28), (5, 29), (5, 30), (5, 31), (5, 32), (5,
33), (5, 34), (5, 35), (6, 35), (6, 34), (6, 33), (6, 32), (6, 31), (6, 30), (6, 29), (6, 28), (7, 28), (7, 29), (7, 30), (7, 31), (7, 32), (7, 33), (7, 34), (7, 35), (8, 35), (8, 34), (8, 33
), (8, 32), (8, 31), (8, 30), (8, 29), (8, 28), (9, 28), (9, 29), (9, 30), (9, 31), (9, 32), (9, 33), (9, 34), (9, 35), (10, 35), (10, 34), (10, 33), (10, 32), (10, 31), (10, 30), (10, 29), (
10, 28), (11, 28), (11, 29), (11, 30), (11, 31), (11, 32), (11, 33), (11, 34), (11, 35), (12, 35), (12, 34), (12, 33), (12, 32), (12, 31), (12, 30), (12, 29), (12, 28), (12, 27), (12, 26), (1
2, 25), (12, 24), (12, 23), (12, 22), (12, 21), (11, 21), (11, 22), (11, 23), (11, 24), (11, 25), (11, 26), (10, 26), (10, 25), (10, 24), (10, 23), (10, 22), (10, 21), (10, 20), (10, 19), (10
, 18), (10, 17), (10, 16), (10, 15), (10, 14), (10, 13), (10, 12), (10, 11), (10, 10), (10, 9), (10, 8), (10, 7), (10, 6), (10, 5), (10, 4), (10, 3), (10, 2), (10, 1), (11, 1), (11, 2), (11,
3), (11, 4), (11, 5), (11, 6), (11, 7), (11, 8), (12, 8), (12, 7), (12, 6), (12, 5), (12, 4), (12, 3), (12, 2), (12, 1), (13, 1), (13, 2), (13, 3), (13, 4), (13, 5), (13, 6), (13, 7), (13, 8)
, (14, 8), (14, 7), (14, 6), (14, 5), (14, 4), (14, 3), (14, 2), (14, 1), (15, 1), (15, 2), (15, 3), (15, 4), (15, 5), (15, 6), (15, 7), (15, 8), (16, 8), (16, 7), (16, 6), (16, 5), (16, 4),
(16, 3), (16, 2), (16, 1), (17, 1), (17, 2), (17, 3), (17, 4), (17, 5), (17, 6), (17, 7), (17, 8), (18, 8), (18, 9)]
DFS path cost: 217
DFS expand node: 384
-------------------------------
BFS path: [(1, 23), (2, 23), (3, 23), (4, 23), (5, 23), (5, 24), (5, 25), (5, 26), (5, 27), (5, 28), (6, 28), (7, 28), (8, 28), (9, 28), (10, 28), (11, 28), (12, 28), (12, 27), (12, 26), (11,
 26), (11, 25), (11, 24), (11, 23), (11, 22), (11, 21), (11, 20), (11, 19), (11, 18), (11, 17), (11, 16), (11, 15), (11, 14), (11, 13), (11, 12), (11, 11), (11, 10), (11, 9), (11, 8), (12, 8)
, (13, 8), (14, 8), (15, 8), (16, 8), (17, 8), (18, 8), (18, 9)]
BFS path cost: 45
BFS expand node: 523
-------------------------------
Greedy Best First path: [(1, 23), (2, 23), (3, 23), (4, 23), (5, 23), (5, 24), (5, 25), (5, 26), (5, 27), (5, 28), (6, 28), (7, 28), (8, 28), (9, 28), (10, 28), (11, 28), (12, 28), (13, 28),
(14, 28), (15, 28), (16, 28), (17, 28), (18, 28), (18, 27), (18, 26), (18, 25), (18, 24), (18, 23), (18, 22), (18, 21), (18, 20), (18, 19), (18, 18), (18, 17), (18, 16), (18, 15), (18, 14), (
18, 13), (18, 12), (17, 12), (17, 13), (17, 14), (16, 14), (16, 15), (15, 15), (15, 16), (15, 17), (15, 18), (14, 18), (14, 19), (14, 20), (14, 21), (13, 21), (12, 21), (11, 21), (11, 20), (1
1, 19), (11, 18), (11, 17), (11, 16), (11, 15), (11, 14), (11, 13), (11, 12), (11, 11), (11, 10), (11, 9), (11, 8), (12, 8), (13, 8), (14, 8), (15, 8), (16, 8), (17, 8), (18, 8), (18, 9)]
Greedy Best First path cost: 75
Greedy Best First expand node: 153
-------------------------------
A* path: [(1, 23), (2, 23), (3, 23), (4, 23), (4, 24), (4, 25), (4, 26), (4, 27), (5, 27), (5, 28), (6, 28), (7, 28), (8, 28), (9, 28), (10, 28), (11, 28), (12, 28), (12, 27), (12, 26), (12,
25), (12, 24), (12, 23), (11, 23), (11, 22), (11, 21), (11, 20), (11, 19), (11, 18), (11, 17), (11, 16), (11, 15), (11, 14), (11, 13), (11, 12), (11, 11), (11, 10), (11, 9), (11, 8), (12, 8),
 (13, 8), (14, 8), (15, 8), (16, 8), (17, 8), (18, 8), (18, 9)]
A* path cost: 45
A* expand node: 237
-------------------------------
```

1.DFS result summary

	mediumMaze	bigMaze	openMaze
Solution Cost	116	240	217
Expand nodes	166	318	384

Medium maze path:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%         % %           %     %           %           %  ...%
% %%%% %% % % %%%%% %%%%% %%% % %%%% %% %%% %% %% %%% % %.% %
% %     %   %     %       %   %       %     %   % % %   %.% %
% % %%% % %%%% %% %%%%%%%%% %%%%% %%% % %%% % % % % %%%%%.% %
% %   %   %       %       %       %   %   % % % %     %...% %
% % %%% %%% %%% %%% % %%% %%%% %% % %%% % %%% % %%% %%%.%%% %
%   %     %     %   % %         % % %   %     %   % %...% % %
% %%% %%% %%%%% %%%%% % %%% %%% %%%%% %%%%%% %%%% %%%.%%% % %
%     %       %         % % %         %...%     %  ...%   % %
% %%%%% %%%%% %%%%% %%% % % % %%%%%%%%%.%.% %%% %%%.%%% %%% %
% %   % %   %         % %   % %     %  .%.% %.......    %   %
% % %%% % %%% %%%%% % % %%%%% % % %%% %.%.%%%.%%%%% %%%%% %%%
%   %   % %       % % %   %   % %   % %.%.....%     %   % % %
% %%% %%% % % %%% %%% %%% % %%% %%% % %.%%%%%%%%% %%% % % % %
% %   %   % %...%   % %   % %     %   %...%     % %   %   % %
%...%%% % % %.%.% % % % %%% % %%%%%%% %%%.% %%% % % %%%%%%% %
%.%...% %   %.%...%   % %    .............% %   % % %       %
%.%%%.%%% %%%.%%%.% %%% % %%%.% %%%%%% %%%% % %%%%% % % %%% %
%...%.........%...% %   % %...%         %   % %     % % %   %
%%%.%%% % %%%%%.%%%%% %%%%%.%%% %%%%%%% % %%% % %%%%%%% % %%%
%P..  %       %.............%         %     %           %   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Big maze Path:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%P....  %.......%...% %     %   %   %   %    .........% %   %.........%.....%   %
%   %.%%%.% %%%.%.%.% % % %%% % % % % %%% %%%.%% %%%%.% % % %.% %%%%%.%.%%%.%%% %
%   %.....% % %...%.%   %   % % % % % %   %   % %.%     %.%   %  .%     %...% %...  %
%%% %%%% %% % %  %%.%% %% % %%% % % % % %%% %.%%%%% %.% %%%%%.%...% % %%% % %.% %
%   % %     %     %.....% %   %   % % %     %.....% %.%     %...%.......%   %.% %
%   % % % %   % % %.....% % % %%% % % %%%%% %%%%%.% %.%%%%% % % % %%% %.%%% %.%%%
% %   % % %   % % %.... %   %     % % ......%.....% %.....%...% %     %...%  ...%
% %%% % %%% % %%% % %..%% %%% %%% % %%.%%%%.%.%%%%% %.....%.%...%% %% %%%.% %%%.%
% %       % % %   % %...% %.....% % ... %  ...%      ...  %.%  .....% %...% %...%
% % %%%%% % % % %%% % %.%%%.%%%.% %%....% %%%%% %%% %%%.%%%.% %%%%%.%%%.%%% %.% %
%   %   %   % %     % %.....%  .........% %   % % %   %.....%     %.....%   %.% %
% % %%% %%%%% % % %%% % % %%% %%%%%%% %%% % %%% % %%% % %% %%%%        %% %%%.% %
% %         % % %   % %   % % %       %   %       % %   %         %     %   %.% %
% % %%% %%%%% % %%% % %%%%% % % %%%%%%%%% %% %%%% % %%%%% %% %%%%%%% %% %%% %.%%%
%   %   %     %   %   %       %         %       %   %   % %   %           % %...%
%%% % %%% %%%%% %%%%%%% %%% %    %% %%% %%%%% % %%% % %       % % %%%% %%%% %%%.%
%       % %           % %   %         %         % %   %   % % % %   %     % %...%
% %%%%%%% % %%%%% %%% % %%% % %%%%%%% %%%%%%%%%%% %%% %%% % % % %%% % %%% % %.%%%
%         %     %   % %   % %   %   %   %   %     %     %   %   % %     % % %...%
%%%%% % %%%%%   %%% % %%% % %%% %%%     % % %%% % % %%% % %%%%%%% %%% %%% % % %.%
%     %   %   %   % % %   %   %           %     %   %   %         %   %   % % %.%
% %%%%%%% % % % % % % % %%% % %%%%% % %%%%%%% %%%%%%% %%%%%%% %%% %%% % %%% % %.%
%       % % %   % % % % %   %     % %       % %   %   %   %     %   % %   % % %.%
%%% %% %% % %%%%% % %%% % % %  %% % %%%% %% %%% % %%%%% % % % % %%% % %%% % %%%.%
%         %   %   %   % % % %   % %       %     %       % % % %   % %   % %    .%
% %%%%    %%% % %%% % % %%% %%% % % %%% % % % %%% % %%%%% % %     % % % % %%%%%.%
%   %     %   % %   %       %   %   %   % % %   % % %     % %     % % % %     %.%
% % %%% %%% %%% %% %%%%%%%%%% %%%%%%% %%% % %%% % % %%% % % % %%%%% %%% %%%%% %.%
% %           %                       %   %     %       %   %               %  .%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

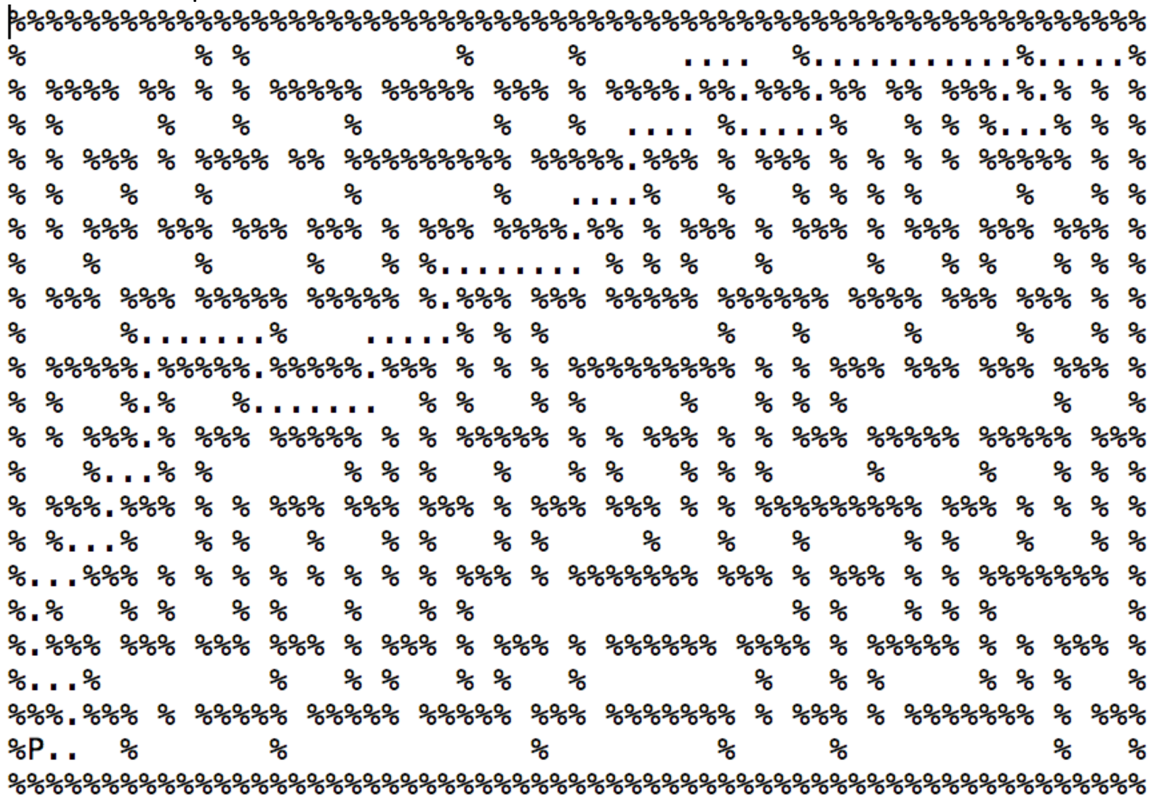
Open maze path:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     %P............%
%                     %.............%
%                     %.............%
%                     %.............%
%                     %.............%
%                     %%%%%%........%
%                          %........%
%                          %........%
%                          %........%
%..........................%........%
%........            ......%........%
%........%%%%%%%%%%%%...............%
%........%                          %
%........%                          %
%........%                          %
%........%                          %
%       ..%                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

2.BFS result summary

	mediumMaze	bigMaze	openMaze
Solution Cost	94	148	45
Expand nodes	608	1255	523

Medium maze path:



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%         % %           %     %     ....  %...........%.....%
% %%%% %% % % %%%%% %%%%% %%% % %%%%.%%.%%%.%% %% %%%.%.% % %
% %     %   %     %       %   %  .... %.....%   % % %...% % %
% % %%% % %%%% %% %%%%%%%%% %%%%%.%%% % %%% % % % % %%%%% % %
% %   %   %       %       %   ....%   %   % % % %     %   % %
% % %%% %%% %%% %%% % %%% %%%%.%% % %%% % %%% % %%% %%% %%% %
%   %     %     %   % %........ % % %   %     %   % %   % % %
% %%% %%% %%%%% %%%%% %.%%% %%% %%%%% %%%%%% %%%% %%% %%% % %
%     %.......%    .....% % %         %   %     %     %   % %
% %%%%%.%%%%%.%%%%%.%%% % % % %%%%%%%%% % % %%% %%% %%% %%% %
% %   %.%   %.......  % %   % %     %   % % %           %   %
% % %%%.% %%% %%%%% % % %%%%% % % %%% % % %%% %%%%% %%%%% %%%
%   %...% %       % % %   %   % %   % % %     %     %   % % %
% %%%.%%% % % %%% %%% %%% % %%% %%% % % %%%%%%%%% %%% % % % %
% %...%   % %   %   % %   % %     %   %   %     % %   %   % %
%...%%% % % % % % % % % %%% % %%%%%%% %%% % %%% % % %%%%%%% %
%.%   % %   % %   %   % %                 % %   % % %       %
%.%%% %%% %%% %%% % %%% % %%% % %%%%%% %%%% % %%%%% % % %%% %
%...%         %   % %   % %   %         %   % %     % % %   %
%%%.%%% % %%%%% %%%%% %%%%% %%% %%%%%%% % %%% % %%%%%%% % %%%
%P..  %       %             %         %     %           %   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Big maze path:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%P....  %.......%...% %     %   %   %   %             % %   %         %     %   %
%   %.%%%.% %%%.%.%.% % % %%% % % % % %%% %%% %% %%%% % % % % % %%%%% % %%% %%% %
%   %.....% % %...%.%   %   % % % % % %   % % %     % %   %   %     %   % %     %
%%% %%%% %% % %  %%.%% %% % %%% % % % % %%% % %%%%% % % %%%%% %   % % %%% % % % %
%   % %     %     %.    % %   %   % % %     %     % % %     %   %       %   % % %
%   % % % %   % % %.    % % % %%% % % %%%%% %%%%% % % %%%%% % % % %%% % %%% % %%%
% %   % % %   % % %...  %   %     % %       %     % %     %   % %     %   %     %
% %%% % %%% % %%% % %. %% %%% %%% % %% %%%% % %%%%% %     % %   %% %% %%% % %%% %
% %       % % %   % %...% %.....% %     %     %.........  % %       % %   % %   %
% % %%%%% % % % %%% % %.%%%.%%%.% %%    % %%%%%.%%% %%%.%%% % %%%%% %%% %%% % % %
%   %   %   % %     % %.....%...        % %   %.% %   %.... %     %     %   % % %
% % %%% %%%%% % % %%% % % %%%.%%%%%%% %%% % %%%.% %%% % %%.%%%%.....   %% %%% % %
% %         % % %   % %   % %.%       %   % ....  % %   % ......  %..   %   % % %
% % %%% %%%%% % %%% % %%%%% %.% %%%%%%%%% %%.%%%% % %%%%% %% %%%%%%%.%% %%% % %%%
%   %   %     %   %   %      .%.........%   ..  %   %   % %   %     ...   % %   %
%%% % %%% %%%%% %%%%%%% %%% %... %% %%%.%%%%%.% %%% % %       % % %%%%.%%%% %%% %
%       % %           % %   %         %.......  % %   %   % % % %   % ....% %   %
% %%%%%%% % %%%%% %%% % %%% % %%%%%%% %%%%%%%%%%% %%% %%% % % % %%% % %%%.% % %%%
%         %     %     % % % %   %   %   %   %     %     %   %   % %     %.% %   %
%%%%% % %%%%%   %%% % %%% % %%% %%%     % % %%% % % %%% % %%%%%%% %%% %%%.% % % %
%     %   %   %   % % %   %   %           %     %   %   %         %   %...% % % %
% %%%%%%% % % % % % % % %%% % %%%%% % %%%%%%% %%%%%%% %%%%%%% %%% %%% %.%%% % % %
%       % % %   % % % % %   %     % %       % %   %   %   %     %   % %...% % % %
%%% %% %% % %%%%% % %%% % % %  %% % %%%% %% %%% % %%%%% % % % % %%% % %%%.% %%% %
%         %   %   %   % % % %   % %       %     %       % % % %   % %   %.%     %
% %%%%    %%% % %%% % % %%% %%% % % %%% % % % %%% % %%%%% % %     % % % %.%%%%% %
%   %     %   % %   %       %   %   %   % % %   % % %     % %     % % % %.....% %
% % %%% %%% %%% %% %%%%%%%%%% %%%%%%% %%% % %%% % % %%% % % % %%%%% %%% %%%%%.% %
% %           %                       %   %     %       %   %               %...%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

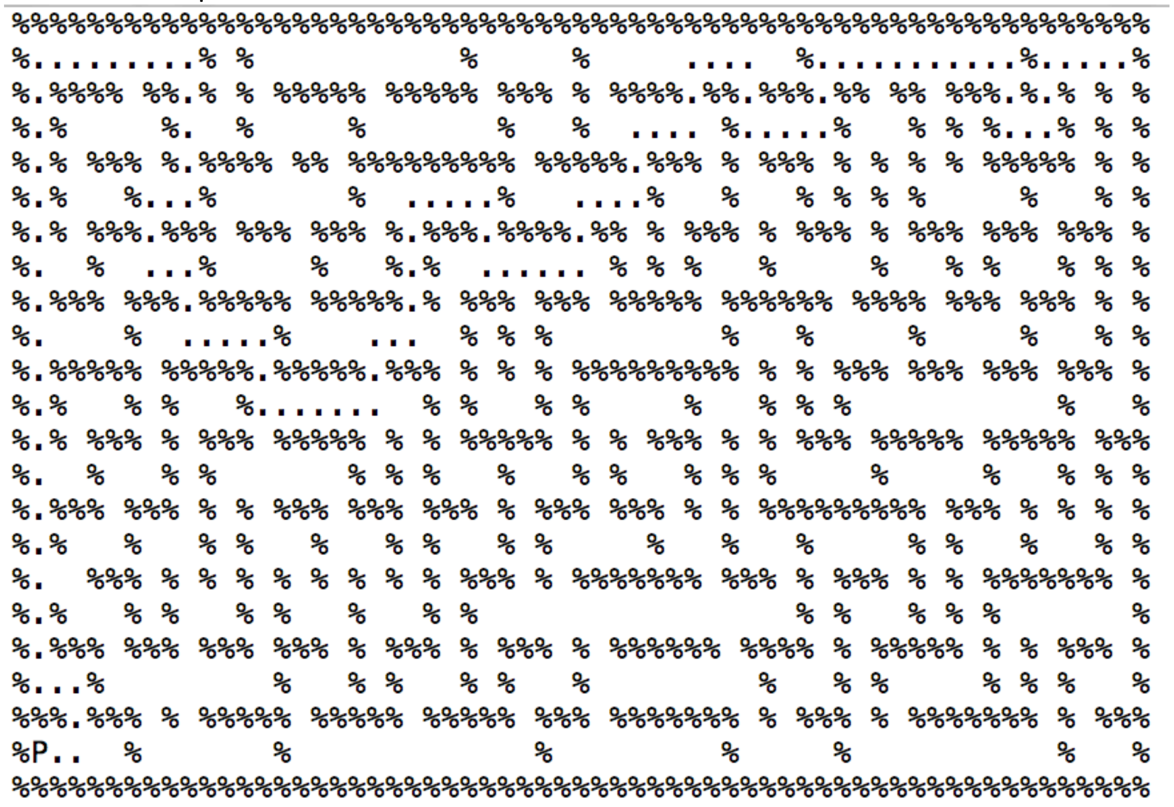
Open maze path:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     %P            %
%                     %.            %
%                     %.            %
%                     %.            %
%                     %......       %
%                     %%%%%%.       %
%                          %.       %
%                          %.       %
%                          %.       %
%                          %.       %
%       ...................%.       %
%       .%%%%%%%%%%%%     ...       %
%       .%                          %
%       .%                          %
%       .%                          %
%       .%                          %
%       .%                          %
%       ..%                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

3.Greedy best-first result summary

	mediumMaze	bigMaze	openMaze
Solution Cost	118	234	75
Expand nodes	133	291	153

Medium maze path:



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%.........% %           %     %     ....  %...........%.....%
%.%%%% %%.% % %%%%% %%%%% %%% % %%%%.%%.%%%.%% %% %%%.%.% % %
%.%     %.  %     %       %   %  .... %.....%   % % %...% % %
%.% %%% %.%%%% %% %%%%%%%%% %%%%%.%%% % %%% % % % % %%%%% % %
%.%   %...%       %  .....%   ....%   %   % % % %     %   % %
%.% %%%.%%% %%% %%% %.%%%.%%%%.%% % %%% % %%% % %%% %%% %%% %
%.  %  ...%     %   %.%  ...... % % %   %     %   % %   % % %
%.%%% %%%.%%%%% %%%%%.% %%% %%% %%%%% %%%%%% %%%% %%% %%% % %
%.    %  .....%    ...  % % %         %   %     %     %   % %
%.%%%%% %%%%%.%%%%%.%%% % % % %%%%%%%%% % % %%% %%% %%% %%% %
%.%   % %   %.......  % %   % %     %   % % %           %   %
%.% %%% % %%% %%%%% % % %%%%% % % %%% % % %%% %%%%% %%%%% %%%
%.  %   % %       % % %   %   % %   % % %     %     %   % % %
%.%%% %%% % % %%% %%% %%% % %%% %%% % % %%%%%%%%% %%% % % % %
%.%   %   % %   %   % %   % %     %   %   %     % %   %   % %
%.  %%% % % % % % % % % %%% % %%%%%%% %%% % %%% % % %%%%%%% %
%.%   % %   % %   %   % %                 % %   % % %       %
%.%%% %%% %%% %%% % %%% % %%% % %%%%%% %%%% % %%%%% % % %%% %
%...%         %   % %   % %   %         %   % %     % % %   %
%%%.%%% % %%%%% %%%%% %%%%% %%% %%%%%%% % %%% % %%%%%%% % %%%
%P..  %       %             %         %     %           %   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Big maze path:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%P      %       %   % %     %   %   %   %             % %   %         %     %   %
%.  % %%% % %%% % % % % % %%% % % % % %%% %%% %% %%%% % % % % % %%%%% % %%% %%% %
%...%     % % %   % %   %   % % % % % %   % % %     % %   %   %     %   % %     %
%%%.%%%% %% % %  %% %% %% % %%% % % % % %%% % %%%%% % % %%%%% %   % % %%% % % % %
%  .% %     %     %     % %   %   % % %     %     % % %     %   %       %   % % %
%  .% % % %   % % %     % % % %%% % % %%%%% %%%%% % % %%%%% % % % %%% % %%% % %%%
% %...% % %...% % %     %   %     % %       %     % %     %   % %     %   %     %
% %%%.% %%%.%.%%% % %  %% %%% %%% % %% %%%% % %%%%% %     % %   %% %% %%% % %%% %
% %  .....%.%.%   % %   % %     % %     %     %           % %       % %   % %   %
% % %%%%%.%.%.% %%% % % %%% %%% % %%    % %%%%% %%% %%% %%% % %%%%% %%% %%% % % %
%   %   %...%.%     % %     %           % %   % % %   %     %     %     %   % % %
% % %%% %%%%%.% % %%% % % %%% %%%%%%% %%% % %%% % %%% % %% %%%%        %% %%% % %
% %         %.% %   % %   % % %       %   %       % %   %         %     %   % % %
% % %%% %%%%%.% %%% % %%%%% % % %%%%%%%%% %% %%%% % %%%%% %% %%%%%%% %% %%% % %%%
%   %   %.....%   %   %       %         %       %   %   % %   %           % %   %
%%% % %%%.%%%%% %%%%%%% %%% %    %% %%% %%%%% % %%% % %       % % %%%% %%%% %%% %
%       %.%           % %   %         %         % %   %   % % % %   %.....% %   %
% %%%%%%%.% %%%%% %%% % %%% % %%%%%%% %%%%%%%%%%% %%% %%% % % % %%% %.%%%.% % %%%
%      ...%     %   % %   % %   %   %   %   %     %     %   %   % %  .  %.% %   %
%%%%% %.%%%%%   %%% % %%% % %%% %%%     % % %%% % % %%% % %%%%%%% %%%.%%%.% % % %
%     %...%...%...% % %   %   %           %     %   %   %         %  .%...% % % %
% %%%%%%%.%.%.%.%.% % % %%% % %%%%% % %%%%%%% %%%%%%% %%%%%%% %%% %%%.%.%%% % % %
%       %.%.%...%.% % % %   %     % %   ....% %...%   %   %...  %   %.%...% % % %
%%% %% %%.%.%%%%%.% %%% % % %  %% % %%%%.%%.%%%.%.%%%%% % %.%.% %%% %.%%%.% %%% %
%        .%...%...%   % % % %   % %    .. %. ...%.      % %.%.%   % %...%.%     %
% %%%%   .%%%.%.%%% % % %%% %%% % % %%%.% %.%.%%%.% %%%%% %.%.    % % %.%.%%%%% %
%   %  ...%...%.%   %       %   %   %...% %.%...%.% %  ...%.%.    % % %.%.....% %
% % %%%.%%%.%%%.%% %%%%%%%%%% %%%%%%%.%%% %.%%%.%.% %%%.%.%.%.%%%%% %%%.%%%%%.% %
% %    .....  %.......................%   %.....%.......%...%...........    %...%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Open maze path:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     %P            %
%                     %.            %
%                     %.            %
%                     %.            %
%                     %......       %
%                     %%%%%%.       %
%                          %.       %
%                          %.       %
%                          %.       %
%                          %.       %
%       ..............     %.       %
%       .%%%%%%%%%%%%.      .       %
%       .%           .      .       %
%       .%        ....      .       %
%       .%     ....         .       %
%       .%    ..            .       %
%       .%  ...             .       %
%       ..% .................       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

4.A* result summary

	mediumMaze	bigMaze	openMaze
Solution Cost	94	148	45
Expand nodes	328	1112	237

Medium maze path:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%         % %           %     %     ....  %...........%.....%
% %%%% %% % % %%%%% %%%%% %%% % %%%%.%%.%%%.%% %% %%%.%.% % %
% %     %   %     %       %   %  .... %.....%   % % %...% % %
% % %%% % %%%% %% %%%%%%%%% %%%%%.%%% % %%% % % % % %%%%% % %
% %   %   %       %       %   ....%   %   % % % %     %   % %
% % %%% %%% %%% %%% % %%% %%%%.%% % %%% % %%% % %%% %%% %%% %
%   %     %     %   % %........ % % %   %     %   % %   % % %
% %%% %%% %%%%% %%%%% %.%%% %%% %%%%% %%%%%% %%%% %%% %%% % %
%     %.......%    .....% % %         %   %     %     %   % %
% %%%%%.%%%%%.%%%%%.%%% % % % %%%%%%%%% % % %%% %%% %%% %%% %
% %   %.%   %.......  % %   % %     %   % % %           %   %
% % %%%.% %%% %%%%% % % %%%%% % % %%% % % %%% %%%%% %%%%% %%%
%   %...% %       % % %   %   % %   % % %     %     %   % % %
% %%%.%%% % % %%% %%% %%% % %%% %%% % % %%%%%%%%% %%% % % % %
% %...%   % %   %   % %   % %     %   %   %     % %   %   % %
%...%%% % % % % % % % % %%% % %%%%%%% %%% % %%% % % %%%%%%% %
%.%   % %   % %   %   % %                 % %   % % %       %
%.%%% %%% %%% %%% % %%% % %%% % %%%%%% %%%% % %%%%% % % %%% %
%...%         %   % %   % %   %         %   % %     % % %   %
%%%.%%% % %%%%% %%%%% %%%%% %%% %%%%%%% % %%% % %%%%%%% % %%%
%P..  %       %             %         %     %           %   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Big maze path:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%P....  %.......%...% %     %   %   %   %             % %   %         %     %   %
%   %.%%%.% %%%.%.%.% % % %%% % % % % %%% %%% %% %%%% % % % % % %%%%% % %%% %%% %
%   %.....% % %...%.%   %   % % % % % %   % % %     % %   %   %     %   % %     %
%%% %%%% %% % %  %%.%% %% % %%% % % % % %%% % %%%%% % % %%%%% %   % % %%% % % % %
%   % %     %     %.    % %   %   % % %     %     % % %     %   %       %   % % %
%   % % % %   % % %.    % % % %%% % % %%%%% %%%%% % % %%%%% % % % %%% % %%% % %%%
% %   % % %   % % %...  %   %     % %       %     % %     %   % %     %   %     %
% %%% % %%% % %%% % %. %% %%% %%% % %% %%%% % %%%%% %     % %   %% %% %%% % %%% %
% %       % % %   % %...% %.....% %     %     %.........  % %       % %   % %   %
% % %%%%% % % % %%% % %.%%%.%%%.% %%    % %%%%%.%%% %%%.%%% % %%%%% %%% %%% % % %
%   %   %   % %     % %.....%...        % %   %.% %   %.... %     %     %   % % %
% % %%% %%%%% % % %%% % % %%%.%%%%%%% %%% % %%%.% %%% % %%.%%%%  ...   %% %%% % %
% %         % % %   % %   % %.%       %   % ....  % %   % ........%..   %   % % %
% % %%% %%%%% % %%% % %%%%% %.% %%%%%%%%% %%.%%%% % %%%%% %% %%%%%%%.%% %%% % %%%
%   %   %     %   %   %      .%.........%   ..  %   %   % %   %     ...   % %   %
%%% % %%% %%%%% %%%%%%% %%% %... %% %%%.%%%%%.% %%% % %       % % %%%%.%%%% %%% %
%       % %           % %   %         %.......  % %   %   % % % %   % ....% %   %
% %%%%%%% % %%%%% %%% % %%% % %%%%%%% %%%%%%%%%%% %%% %%% % % % %%% % %%%.% % %%%
%         %     %   % %   % %   %   %   %   %     %     %   %   % %     %.% %   %
%%%%% % %%%%%   %%% % %%% % %%% %%%     % % %%% % % %%% % %%%%%%% %%% %%%.% % % %
%     %   %   %   % % %   %   %           %     %   %   %         %   %...% % % %
% %%%%%%% % % % % % % % %%% % %%%%% % %%%%%%% %%%%%%% %%%%%%% %%% %%% %.%%% % % %
%       % % %   % % % % %   %     % %       % %   %   %   %     %   %   %...% % % %
%%% %% %% % %%%%% % %%% % % %  %% % %%%% %% %%% % %%%%% % % % % %%% % %%%.% %%% %
%         %   %   %   % % % %   % %       %     %       % % % %   % %   %.%     %
% %%%%    %%% % %%% % % %%% %%% % % %%% % % % %%% % %%%%% % %     % % % %.%%%%% %
%   %     %   % %   %       %   %   %   % % %   % % %     % %     % % % %.....% %
% % %%% %%% %%% %% %%%%%%%%%% %%%%%%% %%% % %%% % % %%% % % % %%%%% %%% %%%%%.% %
% %           %                       %   %     %       %   %               %...%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Open maze path:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     %P            %
%                     %.            %
%                     %.            %
%                     %.....        %
%                     %    ..       %
%                     %%%%%%.       %
%                          %.       %
%                          %.       %
%                          %.       %
%                          %.       %
%       ................   %.       %
%       .%%%%%%%%%%%%  ......       %
%       .%                          %
%       .%                          %
%       .%                          %
%       .%                          %
%       .%                          %
%       ..%                         %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

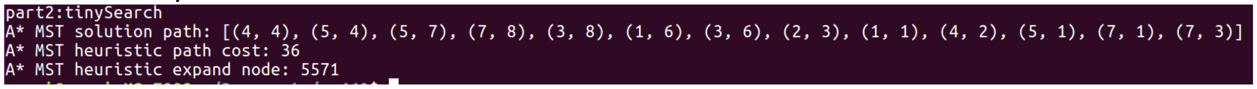
Part 1.2 Implementation Overview

State Transition:
First of all, I load the maze into a double lists, find the start and a list of goal point. Then create mult function, which takes in maze, start point, and a list of end points, return the path, path cost and the total number of expand node.

Algorithm:
First I calculate the shortest distance between each pair of goal position and start position, pushing them in a priority queue so that I can extract smallest at each time later. Then, I create another object PATH to store the path from starting point and visited goals, and PATH also including the total cost of current path and priority as its attributes. The priority is the output of heuristic method, which I will discuss later. I also store PATH in a priority queue, which is sorted using element's priority. Then I can start executing the search procedure: first pop current PATH from priority queue, if its length equals the lengths of goals, it means I have already visited every goal and should return. Else, I create minimum spanning tree (MST) for current visited path. More specifically, its MST connects all goal node that haven't been visited. From this MST I can also calculate the sum of all its tree edges. Then I try to connect current path to every potential goal state that hasn't been visited, by running A* algorithm in part 1.1 from the last node of current path to potential goal state. I use A* because it can guarantee to generate optimal path. Then I can use return path and expand count returned from A* to update the current node, including extending current path, adding total cost of path. Remember we also need to generate an evaluation function to sort the priority queue.$f(n) = g(n) + h(n)$, where $g(n)$ is the path cost so far, and my heuristic function is $h(n) = \sum_{MST} edge + D(last\ visited\ node,\ n)$, which n denote the potential next goal state path would traverse. Then, push the new generated path into priority queue. Keep doing these procedures until we visited every goal state.

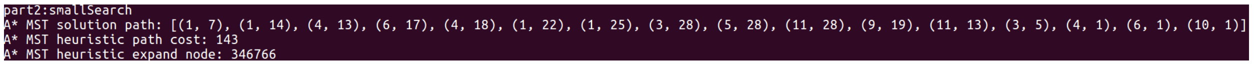
About heuristic and its admissibility:
I claim that my heuristic function $h(n) = \sum_{MST} edge + D(last\ visited\ node,\ n)$ is admissible. First suppose that the pacman traveled the exact math specified by the minimum spanning tree. Because the heuristic relies on the distances calculated using BFS, which are optimal, the path that pacman travels must be greater than the path specified by the minimum spanning tree.
Now suppose there exists a path that pacman can take through all goal nodes and that is less than the sum of weights of the minimum spanning tree. Because pacman reaches every goal node in the graph, its route can be considered as a spanning tree. Therefore, there exists a spanning tree such that the sum of the weight of all edges is less than that of the minimum spanning tree. Which leads to a contraction. Therefore, the minimum spanning tree is less than the pacman true path cost and the heuristic is admissible.

Result for tinySearch:



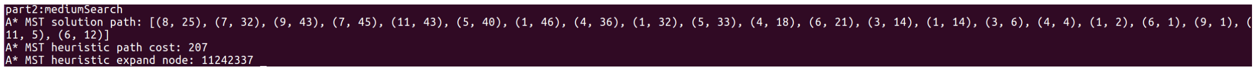
```
part2:tinySearch
A* MST solution path: [(4, 4), (5, 4), (5, 7), (7, 8), (3, 8), (1, 6), (3, 6), (2, 3), (1, 1), (4, 2), (5, 1), (7, 1), (7, 3)]
A* MST heuristic path cost: 36
A* MST heuristic expand node: 5571
```

Result for smallSearch:



```
part2:smallSearch
A* MST solution path: [(1, 7), (1, 14), (4, 13), (6, 17), (4, 18), (1, 22), (1, 25), (3, 28), (5, 28), (11, 28), (9, 19), (11, 13), (3, 5), (4, 1), (6, 1), (10, 1)]
A* MST heuristic path cost: 143
A* MST heuristic expand node: 346766
```

Result for mediumSearch:

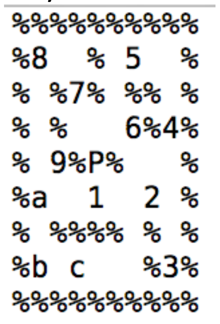


```
part2:mediumSearch
A* MST solution path: [(8, 25), (7, 32), (9, 43), (7, 45), (11, 43), (5, 40), (1, 46), (4, 36), (1, 32), (5, 33), (4, 18), (6, 21), (3, 14), (1, 14), (3, 6), (4, 4), (1, 2), (6, 1), (9, 1), (
11, 5), (6, 12)]
A* MST heuristic path cost: 207
A* MST heuristic expand node: 11242337
```

A* MST heuristic summary

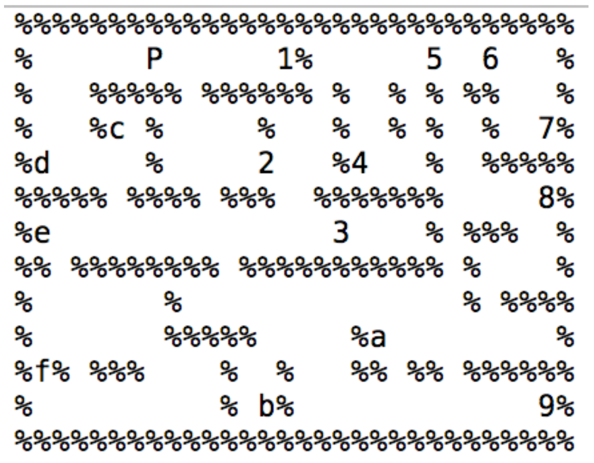
	tinySearch	smallSearch	mediumSearch
Solution Cost	36	143	207
Expand Node	5571	346766	11242337
Time for execute(second)	0.05861090054321	2.9326398727	97.9797420502

Tiny search visited result:



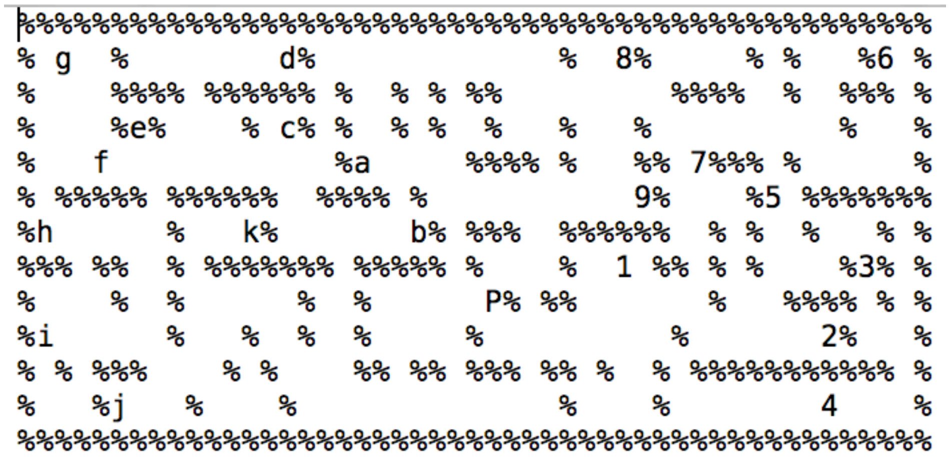
```
%%%%%%%%%%
%8  % 5  %
% %7% %% %
% %   6%4%
% 9%P%   %
%a  1  2 %
% %%%% % %
%b c   %3%
%%%%%%%%%%
```

Small search visited result:



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      P      1%      5  6   %
%   %%%%% %%%%%% %  % % %%   %
%   %c %     %   %  % %  %  7%
%d     %     2   %4   %  %%%%%
%%%%% %%%% %%%  %%%%%%%     8%
%e               3    % %%%  %
%% %%%%%%%% %%%%%%%%%%% %    %
%       %               % %%%%
%       %%%%%     %a         %
%f% %%%    %  %   %% %% %%%%%%
%          % b%             9%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Medium search visited result:



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% g  %        d%             %  8%     % %   %6 %
%    %%%% %%%%%% %  % % %%         %%%%  %  %%% %
%    %e%    % c% %  % %  %   %   %          %   %
%   f            %a     %%%% %   %% 7%%% %      %
% %%%%% %%%%%%  %%%% %           9%    %5 %%%%%%%
%h      %   k%       b% %%%  %%%%%%  % %  %   % %
%%% %%  % %%%%%%% %%%%% %    %  1 %% % %    %3% %
%    %  %      %  %      P% %%       %   %%%% % %
%i      %   %  %  %     %          %       2%   %
% % %%%    % %    %% %% %%% %% %  % %%%%%%%%%%% %
%   %j   %    %              %    %        4    %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Things need to improve:
The time to execute these three searches is much faster than I expected. However, the number of expand node is shockingly large. I think this mainly because I have to find the best path between every two possible goals each time when I try to extend the current path. one idea to optimize this is that I can store the optimal path between every two goal states in advance. However, since the running time is acceptable and due time is coming, I will try to improve this drawback in future.

About source code

The code for part 1.1 is in part1.py, the code for part 1.2 is in part2.py.
To run the code, type python + source code. Notice that part2.py also importing part1.py so when tries to run the part2.py, one can comment out the function in main part1.py for clarity.

Work Contribution Summary

Ruoxi Yang: Do everything.