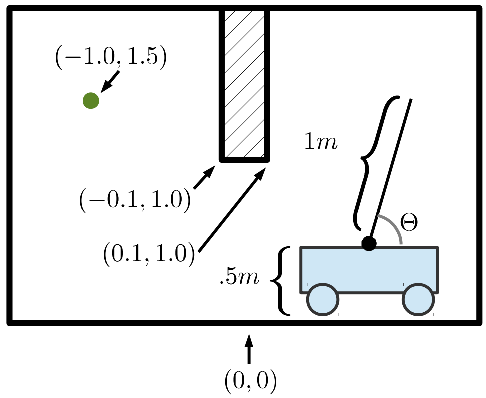
Name: ______________________________

CS354 HW#2, Fall 2019

1. The figure below illustrates a robotic cart. The cart can move to the left or right, and can change the angle of the attached pole. The angle of the pole is indicated by θ, where $\theta = 0$ when the pole is rotated all the way to the right and $\theta = \pi$ when the pole is rotated all the way to the left. The green dot is a goal location for the pole end-point and the hashed box is an obstacle.



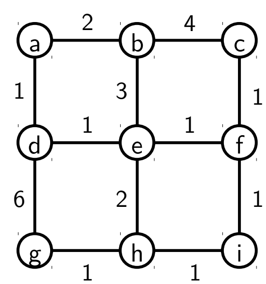
   - Draw the configuration space for this robot, with x on the horizontal axis and θ on the vertical axis. Draw $\mathcal{C}_{obs}$ as a shaded region. (6pts)

   - Draw a valid path in your configuration space from the robot's current configuration to the goal configuration. (2pts)

   - Redraw the configuration space by approximating the robot and the obstacle using minimum bounding circles. Is it still possible to find a path to the goal using this approximation? (3pts)

For each of the questions below, complete the table, then complete the corresponding search method from planning_hw.py. You should be able to check your written answers by executing the corrsponding search. Submit your completed planning_hw.py through Canvas when you submit your answers.

2. Consider the following graph:



Complete the table below to show the state of a DFS search starting at state g with the goal at state a. The tuples in the Frontier column represent Node objects where the first entry is the state and the second entry is the state associated with the parent node. Assume that successors are accessed in alphabetical order. The first three steps are completed for you.

Expansions	Chosen	Frontier	Closed Set
0	–	$\langle g, - \rangle$	–
1	g	$\langle d, g \rangle$, $\langle h, g \rangle$	$\{g\}$
3	h	$\langle d, g \rangle$, $\langle e, h \rangle$, $\langle i, h \rangle$	$\{g, h\}$
3			
4			
5			
6			
7			

What is the final path discovered by DFS? (You should be able to reconstruct it by working backward through the parent entries starting from the point where a is selected from the frontier.)

3. Repeat the previous exercises for a BFS search.

Expansions	Chosen	Frontier	Closed Set
0	–		–
1			
3			
3			
4			

4. Complete the table below to show the state of Dijkstra's algorithm after each expansion. Again the start state is g and the goal state is a. Break priority ties using alphabetical order. I.e. in the case of a tie, state a will selected before state b. The tuples in the Frontier column represent Node objects where the first entry is the state, the second entry is the state associated with the parent node and the third entry is the path cost to reach the state. The first three steps are completed for you.

Expansions	Chosen	Frontier	Closed Set
0	–	$\langle g, -, 0\rangle$	–
1	g	$\langle d, g, 6\rangle$, $\langle h, g, 1\rangle$	$\{g\}$
3	h	$\langle d, g, 6\rangle$, $\langle e, h, 3\rangle$, $\langle i, h, 2\rangle$	$\{g, h\}$
3			
4			
5			
6			
7			
8			

What final path is returned? What is the path cost?

5. Repeat the previous question using an A* search. Use the Manhattan distance to the goal as the heuristic function. For this problem, the fourth value in the Frontier tuples will represent $f(s) = c(s) + h(s)$. The first three expansions are done for you.

Expansions	Chosen	Frontier	Closed Set
0	–	$\langle g, -, 0, 2\rangle$	–
1	g	$\langle d, g, 6, 7\rangle$, $\langle h, g, 1, 4\rangle$	$\{g\}$
3	h	$\langle d, g, 6, 7\rangle$, $\langle e, h, 3, 5\rangle$, $\langle i, h, 2, 6\rangle$	$\{g, h\}$
3			
4			
5			

6. The file rrt.py contains an incomplete implementation of rapidly exploring random trees. Complete the code. You may test your finished implementation using rrt_test.py. You will also need the file spatial_map.py (6pts)