Homework 6

16-311: Introduction to Robotics

Contents

1 Learning Objectives

1. Finish motion planning topics.

2. Practice graph search algorithms.

3. Complete a sample localization problem.

2 Visibility Graph

Here, we will create a visibility graph for a sample workspace with polygonal obstacles. Assume that the start and goal nodes are on the edge of the obstacles. The edges of the image are also obstacles so don't forget those.

Create the visibility graph for the following environment. You may hand-draw this, use an image editor or create a program to accomplish this task.

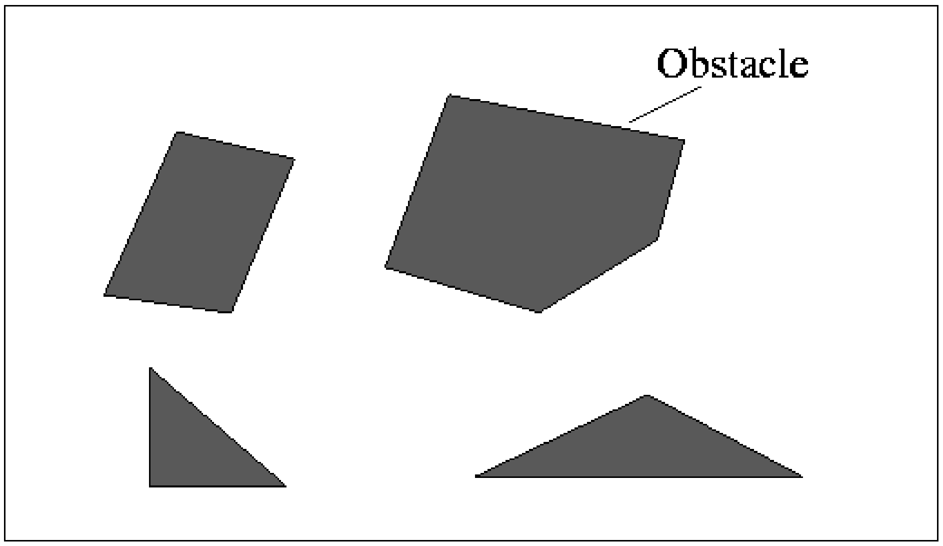

Figure 1: Sample Environment 1.

3 Voronoi Diagram

Using the environment bellow, draw a Voronoi diagram using the Euclidean distance metric (L2). For this metric, you can envision and circle that expands and contracts; when two or more points on the circle touch an obstacle, the center is on the Voronoi diagram.

You can draw this image by hand, use an image editor or write a program that accomplishes this task.

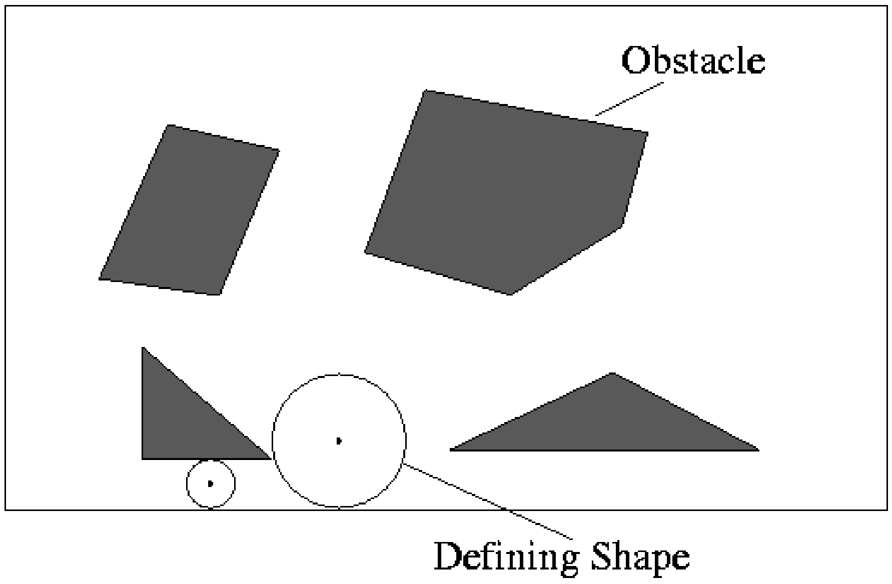

Figure 2: Sample Environment 1 with circle.

4 Graph Search

In this section, we will use breath-first search, depth-first search and A* to find a path from start to goal.

4.1 Search the Graph

First we want to compare searching a graph with Depth First and Breath First Searches. The image below shows our graph, a series of vertices (nodes) and directional edges (arrows). First off, we want to know if A is in our graph. H is our start and A is our goal, but we want to make sure that A is even possible to get to. Note that this search is uninformed, we don't know anything about the environment. Search through the following graphs using first BFS and then DFS. Visit nodes on the same level in alphabetical order. When you get to A (if it is possible), just stop for this question. Please the letters of the nodes you visit in the order you visit them in ending with A (even if there are nodes you haven't yet visited). Think about the best and worst case run times for both of these methods.

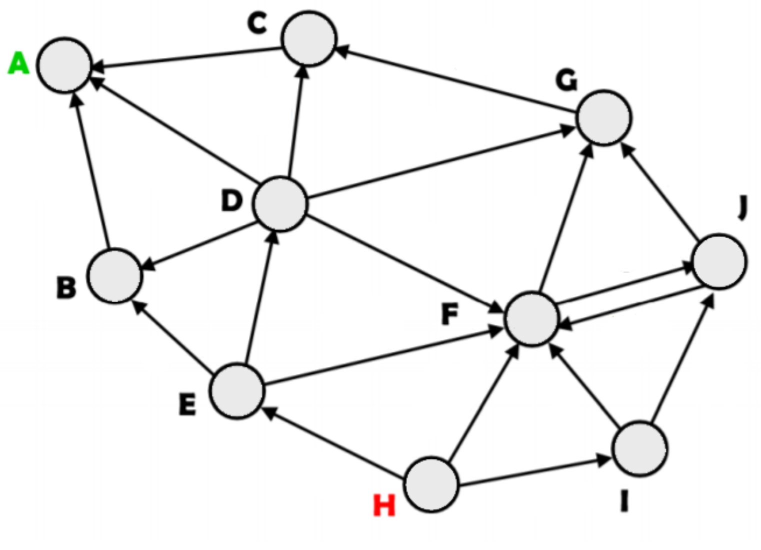

Figure 3: Unweighted graph.

4.2 A Short Path?

Now that we know you can get to A, we want to plan a path from H to A first using DFS. However, we are going to change our rule for getting from one node to the other. Instead of visiting nodes based on the alphabet, we are going to visit nodes based on distance to the node (so you would visit nodes with a smaller edge number first, only using alphabetical ordering to break ties). The distance between nodes is indicated by the number over each arrow. For this problem, we want you to use a very simple hill climbing algorithm (just look for the smallest edge distance and go to that node). Do not keep track of your past decisions and compare them to what you have now (best first). When you get to A, stop. Please list the path you took from H to A. Think about if this was the shortest distance with respect to path length.

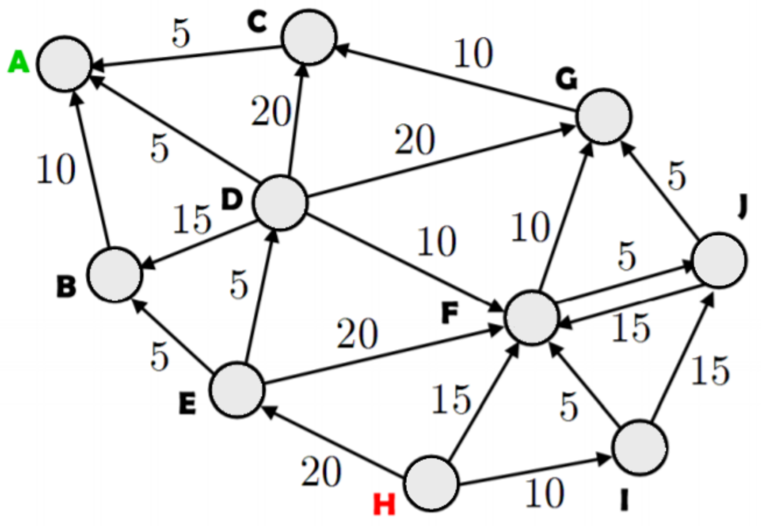

Figure 4: Graph with edge weights.

4.3 A Shorter Path

Finally, we are going to perform an informed search that is guarantied to find the shortest path if we have an admissible and consistent heuristic.

Again, the start is H and the goal is A. The actual cost of each path segment is represented by the number above each arrow. The heuristic value of each node is the number in the center of the circles representing each node.

Please find a path from start to goal using A*. Write a list of nodes that were expanded in the order that they were expanded (i.e. starting at H and ending at A). Also include the path you found.

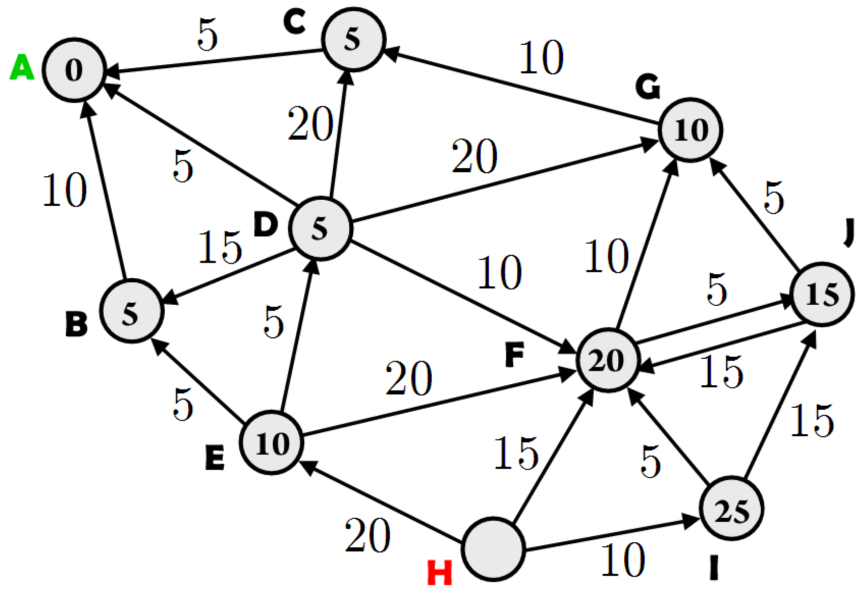

Figure 5: Sample graph.

5 Localization

When localizing, we use sensor information to supplement odometry predictions about where we have traveled. In combination with a known map, we can use these two predictors to make a probabilistic guess of where we are even if we don't know where we start. Here is a video using odometry information and lidar data to predict where the robot is in the world: `https://www.youtube.com/watch?v=1FO009VSAfc`.

For this problem, we want you to simulate a robot's prediction of where it is in this map. The map is an overhead view of a building with two hallways. Assume that the edges of this map are walls. You cannot drive through walls. Demonstrate areas of high probability for the robot's location by putting dots on the map. This is very approximate. There is no correct number of dots or exact placements.

Here is the image of the robot's prediction of where it is at the start (note that it has no strong prediction of where it is in the grid so the dots are just a uniform distribution with a minimum distance from the robot's radius). You do not know the robot's starting orientation, either. The robot to the left is just to demonstrate the shape, its location is not outside of the hallways.

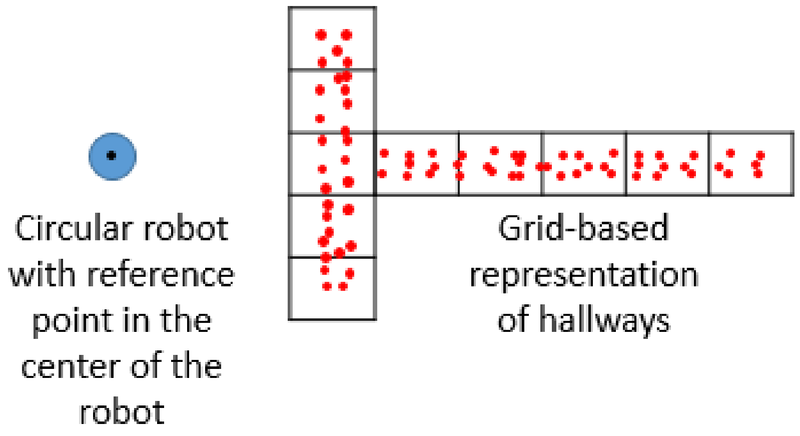
Circular robot with reference point in the center of the robot
Grid-based representation of hallways

Figure 6: Original robot prediction.

Now draw (by hand, using an image editor, etc.) the prediction of where the robot is currently if it has moved (successfully) 4 blocks forward (motion update).

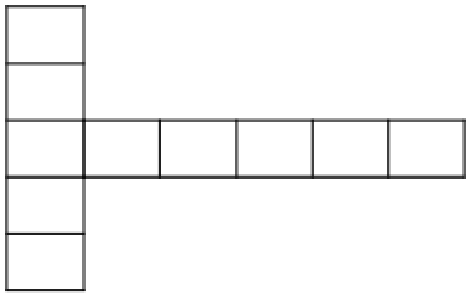

Figure 7: Fill in prediction after motion.

Now draw the prediction of where the robot is currently if it senses a wall directly in front of it 1 block (sensor update).

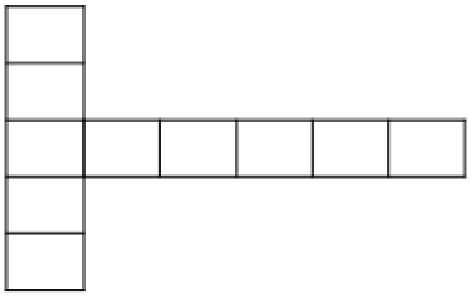

Figure 8: Fill in prediction after sensor reading.

Do we know definitively where the robot is now?

6 What To Submit

Submissions are due on Gradescope by the date specified in the Syllabus.

1. Create a .pdf file with the written answers ALL THE SECTIONS named hw6.pdf.

2. Ensure that your .pdf contain a picture for Part 2, a picture for Part 3, three lists and two paths for Part 4 and two images plus the answer to the question for Part 5.