Algorithmic Robotics
COMP/ELEC/MECH 450/550
Homework 1

DUE: September 5th at the beginning of class (1pm). Submit your answers as a **PDF** to Canvas.

Please read the honor code and the additions described in the course syllabus. Present your work and your work only. You must *explain* all of your answers. Answers without explanation will be given no credit.

1. Exercise 1 **(20 points)** Consider a point robot at q_{start} with the goal of reaching q_{goal} in workspace W which consists of a set of obstacles $WO = \bigcup_{i=1}^{n} WO_i$, where WO_i for all $i \in \{1, 2, \ldots, m\}$ $(m < n)$ is within the radius of $d(q_{start}, q_{goal})$ from q_{goal} and the rest of the obstacles are outside of this radius. What is the maximum number of obstacles the robot will encounter if it uses BUG 1 algorithm? Justify your answer.

2. Exercise 2 **(15 points)** Draw the trajectories produced by Bug 1, Bug 2, and Tangent Bug (with unlimited radius) algorithms for a point robot in the workspace shown in Figure 1.

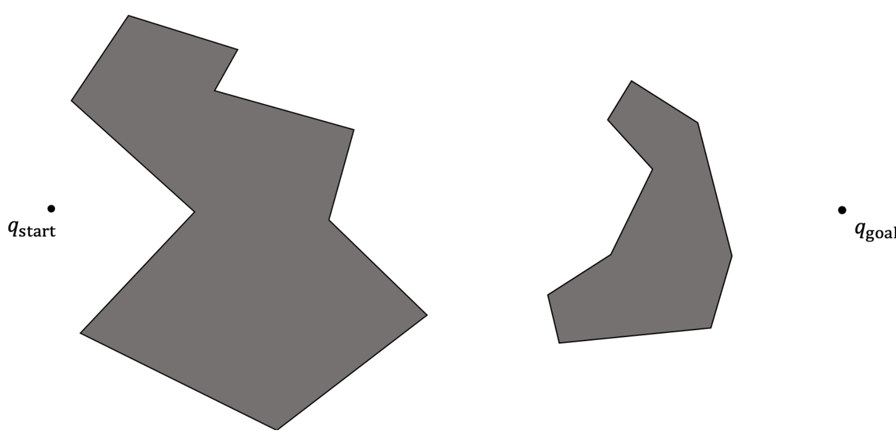

Figure 1: Simple environment.

3. Exercise 3 **(15 points)** Construct an example for which the upper bound of the traveled path for Bug 1 is obtained. How does Bug 2 perform in this example? Explain.

4. Exercise 4 **(20 points)** What is the difference between the Tangent Bug algorithm with zero range detector and Bug 2? Draw an example and explain.

5. Exercise 5 **(10 points)** Give the basics of a convergence proof for the Tangent Bug algorithm. No more than a paragraph is needed.

6. Exercise 6 **(20 points)** Consider the following theorem: Any shortest path between p_{start} and p_{goal} among a set S of disjoint polygonal obstacles is a polygonal path whose inner vertices are vertices of S.

Prove first that a shortest path is a polygonal path. Prove then that the inner vertices of a shortest path τ are vertices of S.

7. Optional Problem **(0 point)** This is an optional problem only for your thought. Please do **NOT** submit your solution to this problem. It will not be graded.

   The construction of the visibility graph makes use of segment-segment intersections. You are given the endpoints to two line segments, A_1B_1 and A_2B_2, in a 2D workspace. The line segments include their endpoints. Provide an algorithm in pseudocode to compute the intersection points of these two line segments, if one exists. Be careful and consider all corner cases. Your algorithm must provide the correct output with every input.

   Hint: There are many ways to represent line segments. Choosing wisely will allow for a shorter and more efficient implementation.