Some tips for doing CS223A problem sets:

- **Print this problem set and fill in your answers in the dedicated white space right below the question statement. Alternatively you may also submit a typewritten writeup.**
- Use abbreviations for trigonometric functions (e.g. $c\theta$ for $\cos(\theta)$, $s_1$ or $s\theta_1$ for $\sin(\theta_1)$) in situations where it would be tedious to repeatedly write $\sin, \cos$, etc.
- Unless instructed otherwise, leave square roots in symbolic form rather than writing out their decimal values.
- If you give a vector as an answer, make sure that you specify what frame it is given in (if it is not clear from context). The same rule applies to rotation and transformation matrices.

1. Consider the RPR planar manipulator in the figure below.

   ![Diagram of RPR planar manipulator](image)

   (a) Assuming joint limits of $0^\circ \leq \theta_1 \leq 360^\circ$, $3 \leq d_2 \leq 6$ and $-90^\circ \leq \theta_3 \leq 90^\circ$, sketch the reachable workspace of this arm. Be sure to label all appropriate dimensions on your sketch.
(b) Does this RPR manipulator’s workspace have a hole in it? If so, outline them in your sketch and calculate its area in the plane. How can you modify the joint limits to eliminate the hole? If not, explain why.

2. From The End Effector To Joint Coordinates: Remember that forward kinematics (converting joint angles → end-effector position) can always be solved with D-H parameters. But when programming robots to do things, we often have a goal position for the end effector in space, and need to find the joint angles that will put it there. Converting from end-effector position → joint angles is called inverse kinematics. Unlike forward kinematics, inverse kinematics requires a custom solution for each robot using geometry. (And, for some robots, no analytical solution is known). This question will explore inverse kinematics.

Consider the planar 2-link RP manipulator shown here (picture shows configuration where $\theta_1 = 0^\circ$):
(a) You are given the \((x, y)\) location (position only) of the end effector. (The end-effector position is the origin of frame 2 expressed in frame 0.) Give an expression for the joint values which will achieve this. Your answer should define \(\{\theta_1, \theta_2\}\) as functions of the \((x, y)\) position of the end effector. Please use a positive \(d_2\).

(b) Now, repeat part a) for the RPR manipulator from question 1, with the additional requirement that the end-effector is holding a plate of food, which means it must stay flat. In other words, \(z_4\) must be pointed directly right (in the direction of \(-y_0\)). Your answer should define \(\{\theta_1, d_2, \theta_3\}\) as functions of the \((x, y)\) position of the end effector. (The end-effector position is the frame 4 origin, expressed in frame 0.)

(c) Suppose we did not constrain the orientation of \(z_4\) in part b). Is there still only one solution for a given end-effector \((x, y)\) position, or many possible solutions? Briefly explain why.
3. **Task and Joint Space Trajectories**: Our goal now is to use inverse kinematics to create a useful end-effector trajectory, such as a straight line in 3D space. Take the RPR manipulator from question 1. Start with end effector (frame \{4\}) coordinates \(0_x^{ee} = 2\sqrt{3}, 0_y^{ee} = -4\). We would like to move the end effector to a new point \(0_x^{ee} = 0, 0_y^{ee} = -8\) in a straight line.

(a) Give the joint coordinates which will produce those end effector positions.

(b) Find the midpoint (i.e. the average) of your joint coordinates from part a). What is the corresponding end effector position?

(c) Now, find the midpoint of the given end-effector start and end positions. Is this the same point as in part b)? If not, what does that imply?

(d) Briefly describe how you could cause the end-effector to follow a straight line from a point A to a point B. (Assume that your robot arm controller takes joint coordinates as input.)