Homework #2
due Wednesday 1/17/24 11:59pm

This week, we are asking folks to make sure their ROS VM is still working, else re-install again. Otherwise, to keep things a little more manageable over the long weekend and avoid debugging issues, I would like to solve the roadmap problems geometrically/analytically. Please draw your answers carefully/accurately and give the exact expressions (in particular for Problem 3). But no need to code anything.

Problem 1 (Check Linux and ROS) - 16 points:

ROS provides a lot of features for autonomous robot navigation (autonomous driving), in particular building and then planning in maps - it would be nice to leverage those later in the quarter.

Indeed, I am hoping that most of us still have their VM running from last quarter. If so, please only check and confirm (step 3 below). If not, or you don't have ROS2, please install a VM following the handouts (see below).

Note, we will not use ROS for another two weeks. So *please* do not spend more than 20 minutes debugging unexpected errors/issues. And work on other problems while the download/install is happening. This is meant to get the process rolling - not to be a time sink!

If you do run into issues, please post questions on Piazza or ask a TA in person. Full credit if everything is working or you remain stuck after having reached out for help. Just work with us to fix problems next week.

We are using Ubuntu 22.04 Jammy Jellyfish (LTS) and ROS2 Humble Hawksbill. The install happens in three steps:

(a) Install Ubuntu, likely via a virtual machine (VM). Options are:

  (i) Commercial products (VMware or Parallels). I have not personally tested.

  (ii) VirtualBox (see instructions on Canvas). Recommended for Windows. Also runs on older Macs with Intel CPUs. It is popular and well documented, though may be slow.

  (iii) Multipass (see instructions on Canvas). Recommended for Macs. Also runs on Windows Pro, Enterprise, or Education (requires Hyper-V). It has worked very well on my Mac.

(b) Customize the Ubuntu setup and install ROS using the instructions and script on Canvas.

(c) Test, making Atlas dance on screen!

  (i) Download the `atlas_description.zip` file from Canvas *into the VM*.

  (ii) Place under `~/robotws/src` and unzip there.

  (iii) Start a new terminal (should show "`Set up ROS2 (local) for ~/robotws`") and build using: (note, this is *NOT* in `src`!)

```
cd ~/robotws
colcon build --symlink-install
source ~/robotws/install/setup.bash
```

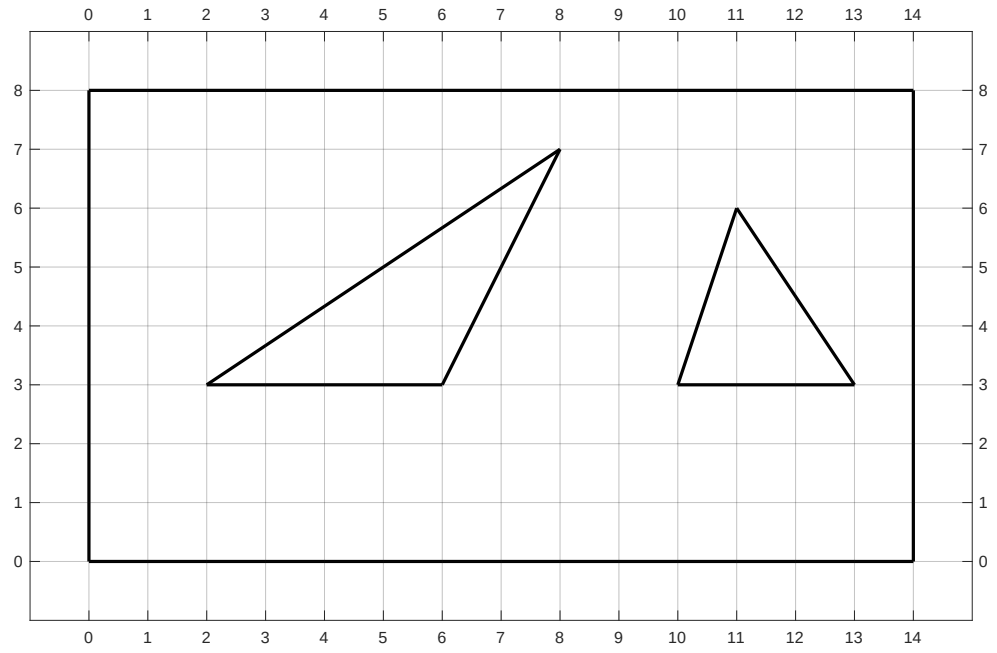
  (iv) Run using the command

```
ros2 launch atlas_description viewatlas.launch.py
```

If everything works, just let us know. Else report any issues you encounter.

Problem 2 (Trapezoidal Cell Decomposition) - 20 points:

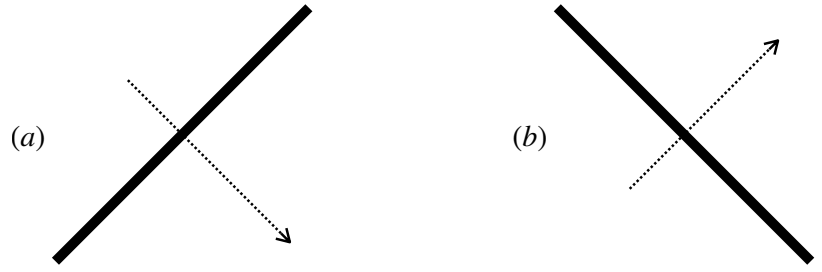
Consider the following room.



Please use a trapezoidal cell decomposition to create a roadmap for the room. We will do this for two sweep angles. In both cases, feel free to draw this by hand. But please clearly mark

- the cell boundaries,
- the graph's nodes and edges.

While the sweep angle could be chosen randomly, let's keep things consistent and sweep a diagonal ($\pm 45°$ slope) line (a) from top-left to bottom-right and (b) from bottom-left to top-right.

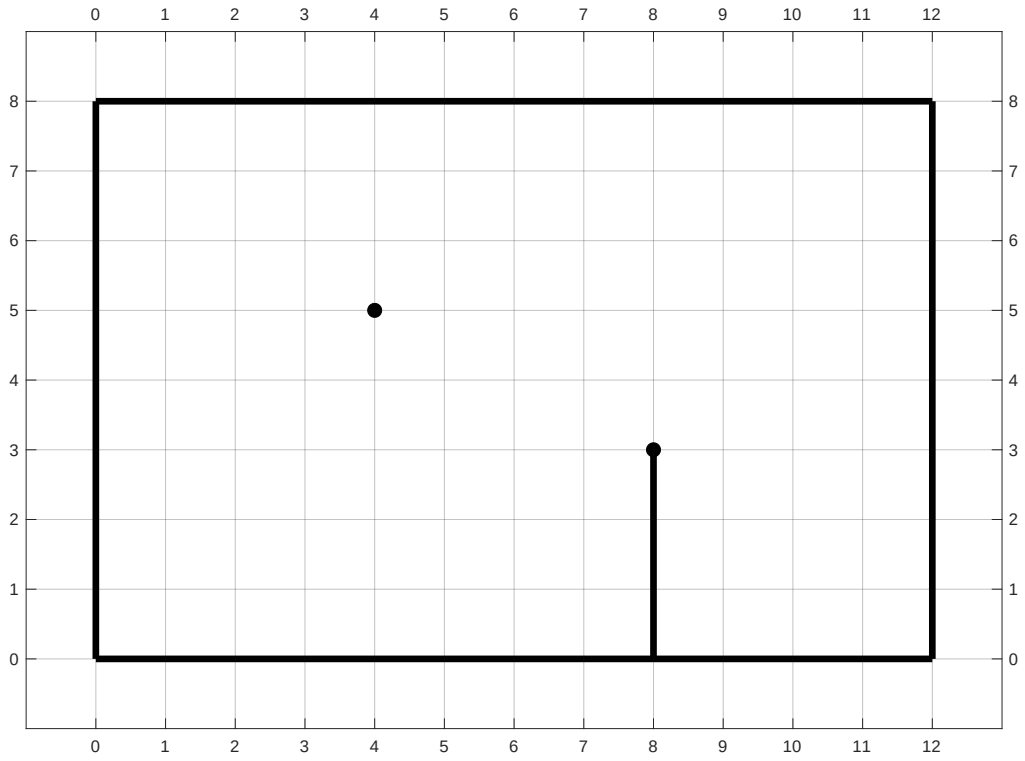


For both cases, please submit your drawings, as well as the number of

- cells,
- nodes in the graph, and
- edges in the graph.

Problem 3 (Generalized Voronoi Diagram) - 20 points:

Please create a roadmap using a generalized Voronoi diagram. Consider the room with the dimensions/locations as indicated in the figure. Assume the points have zero dimension (radius).



Please

(a) Carefully draw the Generalized Voronoi Diagram (GVD) for a maximum clearance roadmap and submit.

(b) How many *internal* intersections (*excluding* the room border) are in the roadmap?

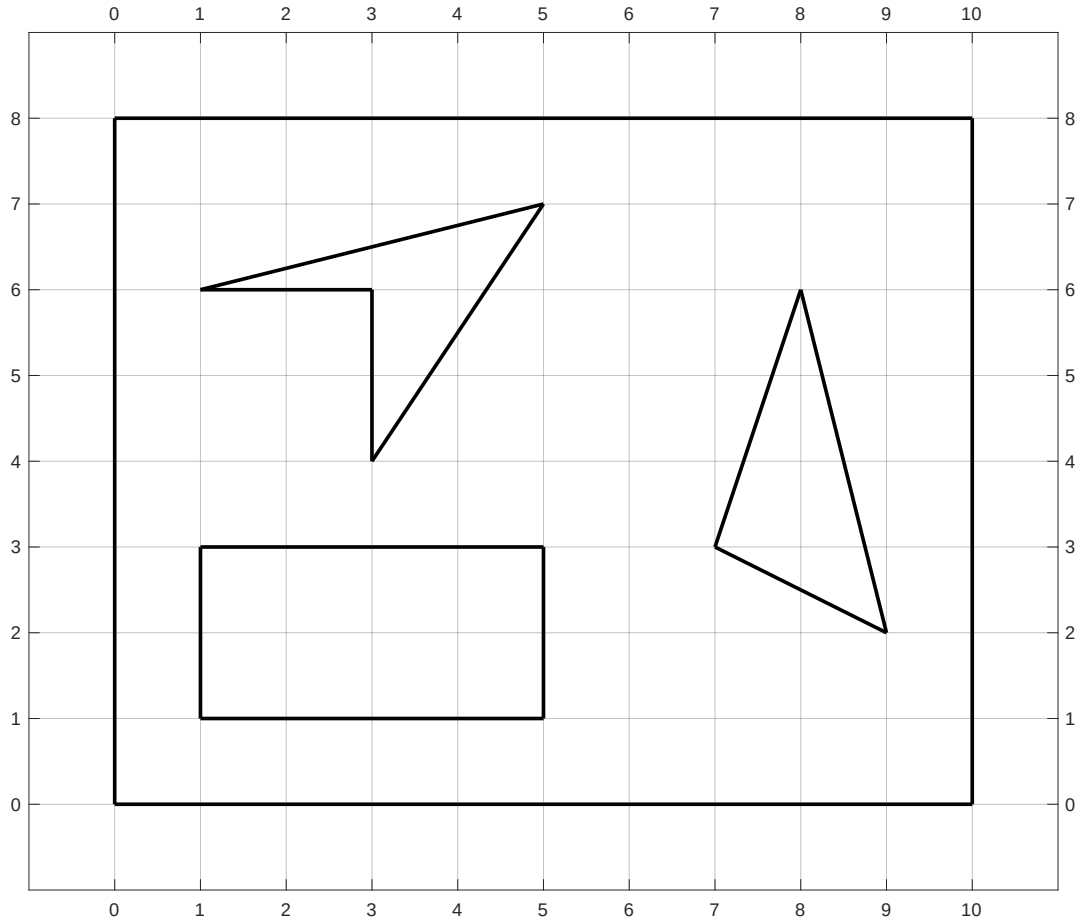
(c) How many total nodes (*including* at the border) are in the graph of the roadmap?

(d) How many edges are in the graph (paths between nodes)?

(e) Please give the equations for each edge/path (something like $\frac{1}{2}(y-2)^2 = (x-4)$).

Problem 4 (Visibility Roadmap) - 20 points:

Please create a shortest-path roadmap for the following room.



Please report/show the reduced visibility roadmap, that is the roadmap with unnecessary nodes/edges removed.

Please

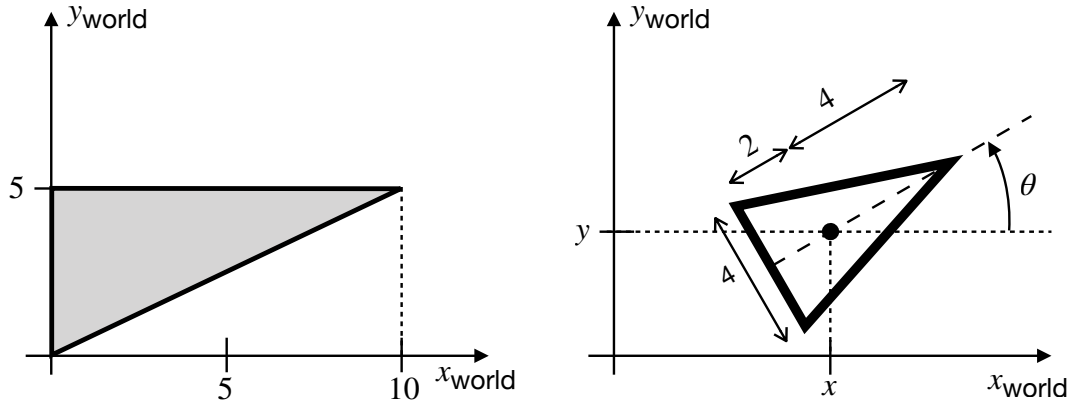
(a) Carefully draw the final (reduced) roadmap.

(b) How many nodes are in the final roadmap's graph?

(c) How many edges are in the final roadmap's graph?

Problem 5 (C-Space Obstacle) - 20 points:

Let's start thinking about higher degree of freedom systems and start by exploring C-Space obstacles. Consider the following planar obstacle (left) and robot (right).



The obstacle is a triangle with the dimensions shown. The bottom left corner is coincident with the world's origin. The robot is a isosceles triangle, height 6 units, base 4 units. The robot's position is defined as the location of the marked dot, on the centerline 2 units from the base. The robot's orientation is defined as the angle of the centerline w.r.t. the world's x axis (about the positive z axis).

Let's see what the obstacle looks like in 3 dimensional C-space (x, y, θ) coordinates of the robot. For simplicity, we'll draw the obstacle only in the (x, y) plane for specific θ angles.

More specifically, please draw the boundary of the obstacle in the (x,y) C-space for the given, fixed orientation θ of the robot. This should be accurate and to scale, i.e. please give a scale. But we are *not* asking for the exact coordinates of the corners. Repeat for the three cases:

(a) $\theta = 0°$

(b) $\theta = 90°$

(c) $\theta = 180°$

Problem 6 (Time Spent) - 4 points:

Approximately how much time did you spend on this homework?

And, if you are handing in late, would you prefer to use penalty-free late hours (default) or take a 10 points/day penalty (we are happy to change later).