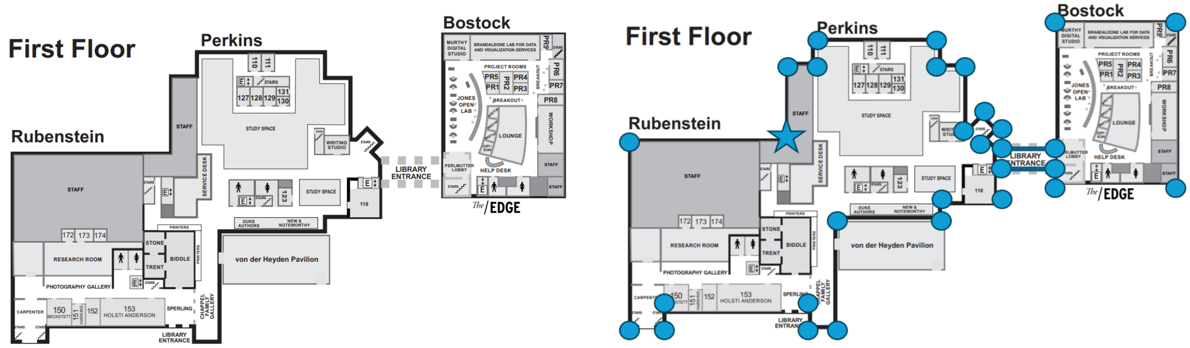
CS230 EMD (Graph Applications in Robotics) Assignment

Assignment Due LDoC 04/24

1 Gradescope Assignment



(a) Layout of Rubenstein/Perkins/Bostock 1st floor. (b) Corners of the layout.

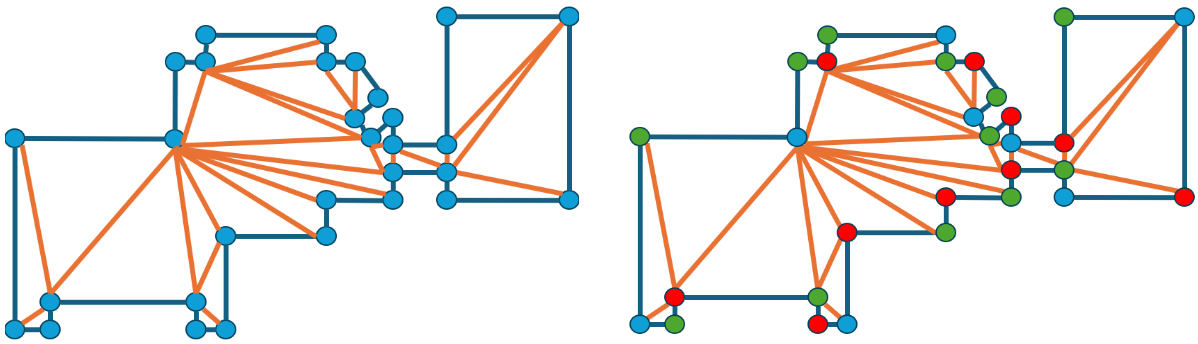
Figure 1: Example of the problem.

We saw in class (briefly) the application of visibility graphs in navigation. In this assignment, we explore another application of visibility graphs, this time *indoors*. Imagine we are guarding a museum/library with a complicated indoor layout (see Figure 1a for the layout of the first floor of Rubenstein/Perkins/Bostock) using surveillance cameras. The abstracted version of the problem is as follows:

- The indoor layout can be represented by a simple polygon. This means (1) every edge is straight, (2) there are no interior holes, and (3) the whole interior is a connected region. Our concrete example of Rubenstein/Perkins/Bostock first floor satisfies these assumptions once we treat the small connector area between Perkins and Bostock as part of the layout (and then imagine that there is currently nothing inside the walls – no bookshelves, no desks, no compartments).
- Surveillance cameras can only be placed at corners of the layout (maybe to not interfere with other things). In other words, if we think of the polygon as a graph, we can only place cameras at vertices (see Figure 1b).
- Cameras can rotate 360 degrees. Therefore, if we place a camera at the star-shaped corner in Figure 1b, that camera covers *most* (but not all of) Rubenstein/Perkins. Of course, cameras cannot capture anywhere that's obstructed by walls, so most of Bostock is out of reach for that camera.

We are interested in **how many cameras are necessary to cover the entire layout so that every inch of the layout is captured by at least one camera**. Of course, the most overkill solution is to put a camera at every corner (convince yourself that this does capture every part of the layout!). On the other hand, solving for the exact minimum is usually too hard. But through some graph theory, we can prove a better upper bound. Here is a step-by-step recipe:

1. (Triangle.) Assume the input graph is just a triangle. **How many cameras do we need?**
   No justification needed. Just answer a number.
2. (Any polygon layout can be divided into triangles.) Now convince yourself that no matter how complicated our input layout is, we can divide the layout into many triangles without adding any new vertices. See Figure 2a for an illustration of dividing the library layout into triangles. **If the input layout had n edges, how many triangles are there?**
   No need to justify your answer, although you can easily prove that by some high-school level geometry.



(a) Library layout divided into triangles.

(b) A 3-coloring of the layout.

Figure 2: Abstraction of the problem.

3. Now, observe that the abstract graph (with many triangles) is a planar graph. Not only is it a planar graph; in fact, *every vertex in this graph is on the same face* (still see Figure 2a; this face is just the original layout boundaries). **Prove that every planar graph with this additional property is 3-colorable.**
   Hint. This can be proved via induction. Alternatively... there is a much shorter proof, which requires thinking outside the box (or the face).

4. Therefore, consider an arbitrary 3-coloring of the planar graph in Part 3 (in which every vertex is on the same face, and every other face is a triangle). **Prove that for each of the three colors, the set of all vertices mapped to that color is a valid solution to our original problem.**
   Hint. The argument relies on some earlier part.

5. Combining all of the above, argue that **we need *at most* $\lfloor \frac{n}{3} \rfloor$ cameras** if the input layout had n edges.