Guards in Galleries

From galleries to geometry and graph theory, then back again

Anna Lindeberg

November 11, 2024

Imagine that you have been tasked to protect a gallery filled with priceless pieces of art. To do this, you need to place guards that watch every nook and cranny of the place. At the same time, your boss is a bit annoyed with you after you bought a banana duct taped to a wall for a truly extraordinary sum of money, so you really want to appoint as few guards as possible. So, what is the smallest number of guards needed to oversee the whole gallery?

What you have been described above is *The Art Gallery Problem*. In all its simplicity, it can still be used to outline many exciting things within mathematics: from geometry, into discretized problems and graph theory, vertex colorings and much else. Do not worry if none of these words make any sense to you yet – they will, after reading this text! All you need is some curiosity and a high school-level grasp of mathematics. It is also worth noting that The Art Gallery Problem is one of many similar problems; there is of course no restriction to art galleries and the guards might move around or not be guards at all. For example, the task of placing different types of sensors and different settings of robotics give rise to similar types of problems with solid real-world applications. But for me, as a mathematician, the beauty of The Art Gallery Problem lies in that it is easy to grasp, yet still covers a lot of interesting and intricate ideas from vast subareas of mathematics. So let us dive right in!

Floorplans and Visibility to Polygons and Lines

To make The Art Gallery Problem into a well posed mathematical one, we need to clarify two things:

1. What is a floorplan of a gallery, *mathematically speaking*?
2. What do we mean when we say that a guard guards something?

The one-word-answer for the first question is: a *polygon*. Personally, I do believe we need a little bit more than that though – even if you probably did hear about polygons in school. Usually, people think about regular polygons: squares, triangles and hexagons. That's a good start! There are, however, other shapes that are polygons too, such as stars and all sorts of shapes that have not been honored with specific names. In short, polygons are shapes that you can draw on a piece of paper *using only straight lines*. No circles or bendy bits allowed! Technically, we also only care about so-called simple polygons, so if you bring out your ruler with intention of drawing some nice floorplan-polygons, make sure that you also never draw any lines that cross each other. In particular, this means our polygons will never have any holes, which admittedly is a little bit of a restriction but it is one we have to live with. Look at Figure 1, which shows two shapes we don't allow (one with crossing edges, and one with a hole) and, more importantly, a rather reasonable floorplan-polygon

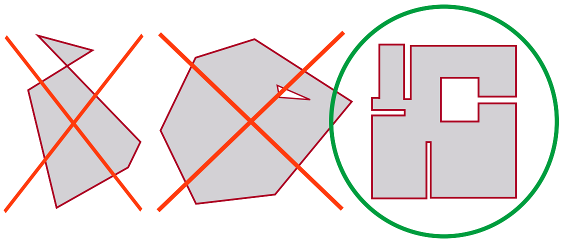

Figure 1: *A floorplan is a polygon: a shape without holes, drawn with straight lines that never cross.*

which we will reuse a couple of times throughout this text.

There is one more thing we need to mention about polygons: *edges* and *corners*. The edges of the polygon are the lines we drew to make the polygon, and the corners are where the edges meet. In the floorplan-polygon in Figure 1 you can count the 26 corners if you want. Easier examples include triangles and squares which have three respectively four corners each.

Onto the next thing! We want to oversee every piece of our floorplan. Now that the floorplan is just a polygon, we may think of placing a guard as picking a point in this polygon. For simplicity, we will assume that every guard is stationary; no guards move around. To compensate for this laziness, we at least make sure that a guard can turn around and look in any direction. What does a guard see then? Well, we say that a point inside the polygon is *visible* from a guard, if we can draw a straight line between the point and the guard's point so that the line never leaves the polygon – see Figure 2 for an example. That last part of the line being contained in the polygon is very important, since it makes sure that guards cannot see through walls.

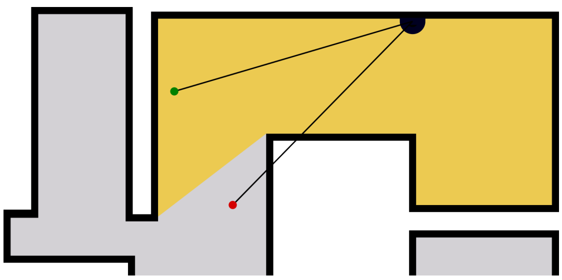

Figure 2: *A guard (big black dot) can see a point (smaller dots) of the floorplan if the straight line between them lies fully within the floorplan itself. The region of points this particular guard sees is indicated in yellow.*

At this stage, we can finally answer the second question we asked: a gallery is guarded by a group of guards if every point in the floorplan is visible from at least one guard. At heart, nothing has changed, but we have formalized the original fuzzy statement into something even mathematicians accept as reasonable. Even more excitingly, we can write down what we want to prove!

Theorem. *You never need more than $n/3$ guards to guard a gallery with n corners.*

Don't be scared by the fancy word "theorem", it is just maths-speech for "a sentence that is true". Instead, be proud that you now have read one[1] that you actually understand! The content of the theorem is also *sort of* an answer to the very first question we asked; what is the minimum number of guards needed? Well, we don't know exactly unless you don't give us more specific information, but can at least say that it **for sure** won't be more than the number of corners of the room divided by three. For example, if you have a gallery with six corners, the theorem tells us that we need at most two guards, since $6/2 = 3$. Actually, if the six-corner-gallery happens to be a hexagon, it suffices with just one guard – but one is less than two, so it's all in line with what the theorem says.

A small remark here is that $n/3$ may not be a whole number. For example, with four corners we get $n = 4$ and consequently need at most $4/3 = 1.333\ldots$ guards. In such cases, it is actually completely fine to round down $n/3$ to the nearest whole number; the theorem is still true.

What will happen now? Well, we will *prove* this theorem! As mentioned earlier, it happens to paint an enthralling story about many things: polygons and geometry, graphs and colorings, how mathematicians use domino effects and more. First stop: how to discretize this problem.

Polygons and Triangles to Graphs

One thing that makes The Art Gallery Problem difficult from a computational point of view, is that there are infinitely many possible placements of a single guard. Even if the floorplan-polygon is finite in size (measured in, say, square meters) we can nudge a guard's placement ever so slightly in some direction, and the points now visible from

[1]Maybe you already know other too; the Pythagorean theorem is a classic few manage to avoid throughout school.

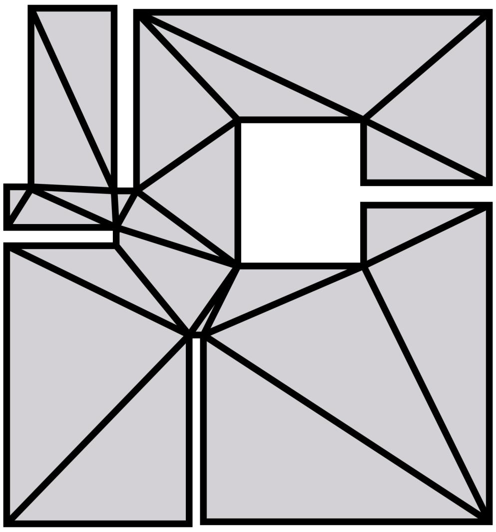

Figure 3: *A triangulation of the art gallery's floorplan, that is, a division of the polygon into a collection of smaller triangles.*

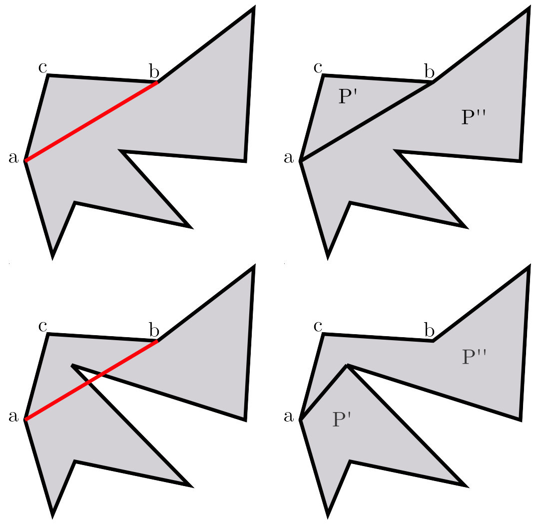

Figure 4: *Triangulating a polygon can by done by drawing diagonals between corners, dividing the polygon into smaller and smaller pieces, repeating the process as necessary.*

the guard may change quite drastically – perhaps the guard suddenly sees around a corner that earlier obstructed the view considerably. To get around this problem, we will do something mathematicians call *discretize*. It means, simply put, to go from something with infinitely many choices (or numbers, or data points, or so) to finitely many. In this case, we will subdivide the floorplan into a bunch of triangles and then consider placements of guards only in the corners of these triangles. This is called *triangulating* a polygon – see Figure 3. In essence, we will draw non-overlapping lines between corners of the polygon so that every small region between lines is a triangle.

A good question to ask here is: can we triangulate every polygon? The answer is yes and the argument for why this is the case goes as follows: if your polygon is a triangle, then it is already triangulated and you don't have to do anything. If your polygon isn't a triangle, then it has at least four corners and you can always pick three consecutive corners in a way such that the straight line between the first and last corners lies at least partially inside the polygon. Look at Figure 4, where two such "diagonals" are shown in red, drawn between corners labeled a and c. If the diagonal lies fully within the polygon, you have split the polygon into two smaller pieces, each with fewer corners than the polygon you started with. If the diagonal at some point leaves the polygon, follow the diagonal from the corner a until you find the first time the diagonal crosses an edge of the polygon. That edge has a corner inside the triangle formed by a, b and c. Instead of using the diagonal between a and c, we draw a *new* diagonal from a to that corner inside the abc-triangle. Then you have managed the same thing once again; you have split the polygon into two smaller pieces and can now repeat the process on those pieces until every piece is a triangle. That is your triangulation.

Hopefully, you're now convinced that we can triangulate every polygon. This is helpful for us, since every point in the polygon now lies in one of the triangles, and every point in a triangle is visible from the corners of that triangle. Therefore, it is possible to place guards only at corners of these triangles and still guard the whole gallery. Since the corners are finitely many, we've discretized our problem! But which corners do we pick? Well, that's the topic of the next section. However, before that, we're going to make one more small mathematical jump, into the world of graph theory.

Let's put it like this: we can still remove some information from the polygon without any problems, and by doing so, we further simplify the problem of guard placement. The only things we need later on, is (1) which corners the polygon has and (2) what the triangles in the triangulation are. Information such as how large different triangles are, exactly where the corners lie and similar actually plays no role in where we will place guards. Why so? Well, when we place a guard at the corner of

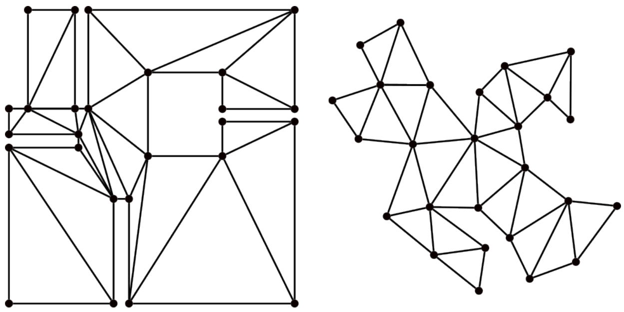

Figure 5: *Two graphs or, really, the same graph drawn in two different ways. The dots are the vertices and the lines are the edges.*

a triangle, the guard will see every point of the triangle *no matter what the triangle looks like*. We do this by creating a *graph* from our triangulated polygon. The graph consists of two things: *vertices* and *edges*. More precisely, we create a graph from our polygon by creating a vertex for each corner in in the polygon, and draw an edge between two vertices if there is a line in the triangulation between the corresponding corners. Figure 5 shows the graph version of the triangulated polygon in Figure 3. In the left version, the correspondence between the graph and the polygon is more or less obvious, but the graph to the right is actually *the same* graph, the dots that represent the vertices have just been shifted somewhat.

We will give the graphs that we obtain from a triangulated polygon a special name: *triangular graphs*, since this sort of graph consist of a bunch of triangles glued together. If not immediately obvious, I believe you can convince yourself that the number of vertices in a triangular graph is the same as the number of corners of the associated polygon.

Graph Colorings

Okay, we have a triangular graph. We will place guards at some of the vertices of that graph. This will, somehow, prove that we never need more guards than a third of the number of corners in the gallery. To do this, we will do something seemingly unrelated, namely, we will assign each vertex a color: blue, red or yellow. While doing so, we will make sure that no two adjacent vertices, i.e. vertices with an edge between them, get the same color. We call this type of coloring of the vertices a *proper coloring*, see Figure 6. If we manage, this means that every triangle in the triangulation

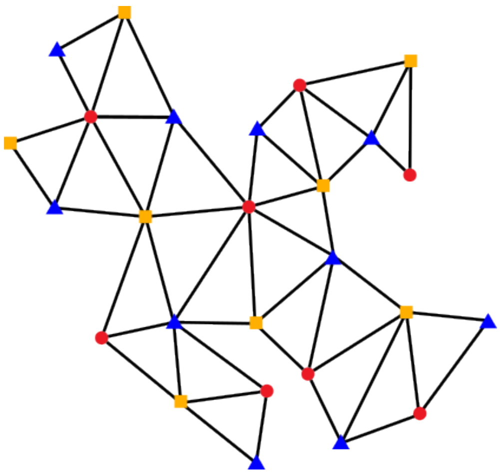

Figure 6: *A proper coloring of the triangular graph from Figure 5.*

have corners with three different colors. In case you happen to be really impatient, I will tell you already now that we will place guards at either all the blue vertices, all the red vertices or all the yellow vertices. But more on that later! First: are we really, really sure that we can color the vertices like this? Probably not, but we will be soon.

We will use *induction*, which is just a fancy way of writing “mathematical domino effect”. It consists of two parts: showing that the first domino brick falls, and showing that if one domino brick falls then so does the next domino brick. In fact, you have already seen it! This was implicitly what we used when we showed that we can triangulate a polygon. If you remember, we first said that a triangle is triangulated – that’s the first brick tilting over – and then showed that if the polygon is larger we can divide it into two parts, triangulate the parts and get a triangulation of the full thing. The triangulation of the smaller parts are, in this simile, the dominos earlier in the row, toppling over the current brick.

For coloring the vertices with blue, red and yellow we also start with a triangle. Here there’s just three vertices, and we can give them one color each. Clearly, no two adjacent vertices have the same color, so this is a proper coloring. Now comes the tricky part: assume that there is some number k so that every triangular graph with at most k vertices can be properly colored. As we just argued, $k = 3$ is one such value of k, but there will soon be others too. Now take any triangular graph with $k+1$ vertices. Since the number of corners in the polygon and the number of vertices in the triangular graph is the same, we have at least four

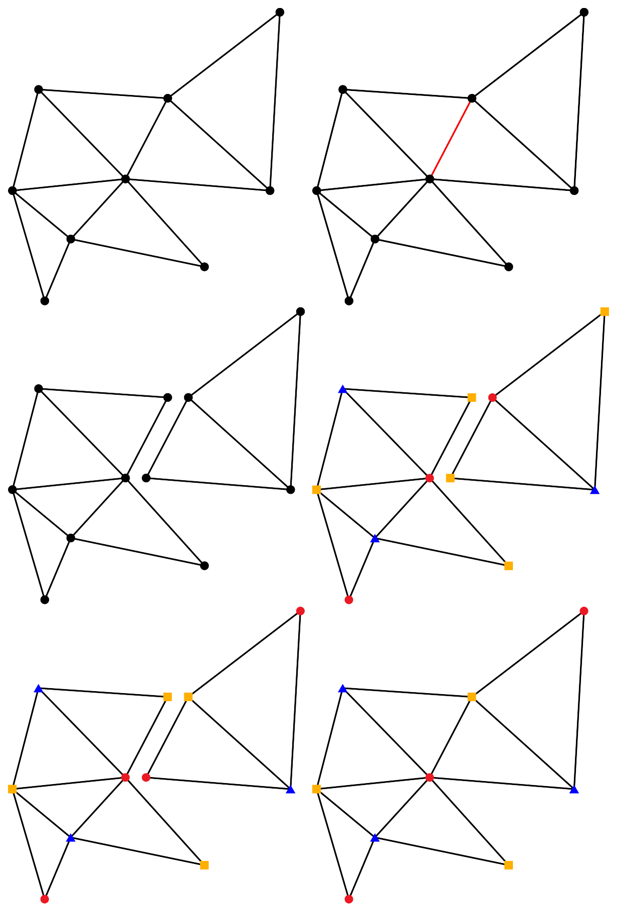

Figure 7: *One can always find an edge between two triangles in a triangular graph (top row) which divides the graph into two smaller triangular graphs that can be properly colored (middle row). Then, one can swap colors of one of the small triangular graphs and glue the graphs back together to get a proper coloring of the large graph (bottom row).*

vertices and at least two triangles in the graph. Therefore, we can find an edge that belongs to two different triangles at the same time and can, so to speak, cut our graph into two parts along that edge – see Figure 7.

The two smaller graphs will be triangular too, and they are both smaller than the graph we started with. In particular they have at most k vertices and there must therefore exist a proper coloring of them – we assumed so! For example, if the big graph has four vertices, then cutting it into two smaller graphs means each part will have three vertices, and those two triangles have proper colorings. We now want to find a proper coloring of the large graph. To do this, start with the colorings of the small ones. If the colors along the edge we used to cut up the graph are the same, we can immediately glue the two smaller colorings together. Otherwise, we can swap colors in one of the smaller graphs so that it does match with the other small graph. For example, we might have to make all blue vertices to red, all red vertices to yellow and all yellow vertices to blue. The coloring you get will still be proper – it's just a way of shifting the names of the colors around, right? In Figure 7 the only thing you need to do is change the yellow vertices to red and vice versa. After potentially swapping colors in one of the graphs, you can always glue the proper colorings of the small graphs together and get a proper coloring of the bigger graph. Voila!

If this was uncomfortably abstract for you, it might help to think in more concrete steps. It was easy to show that a triangle can be properly colored. A triangular graph with four vertices can be cut into two triangles and those triangles can be properly colored and glued together, so a triangular graph with four vertices can also be properly colored. Similarly, if a graph has five vertices, then it can be cut into a triangle and a graph with four vertices (try drawing it!). As just argued, both of those smaller graphs can be properly colored and glued together to a proper coloring of the five vertex graph. Then rinse and repeat! The dominos fall. Neat, is it not?

One particularly interesting thing about vertex colorings is that it is a very, and I really mean **very**, well-studied problem. Its most famous application is probably the *Four Color Theorem* which, in essence, states that every map in the world can be colored so that no two countries (or regions, states, counties, ...) that share a border are drawn with the same color. This was an open problem for almost 200 years, and was one of the first theorems that was proved with the aid of a computer – but all of that could be covered in a text as long as this one.

Putting it all together

At this point, a recap is called for. We want to prove that if we are given a floorplan-polygon with n corners, then it is possible to place no more than $n/3$ (rounded down) guards to guard the whole floorplan. The first step of the proof was to triangulate the polygon, and then transform the triangulation into a triangular graph. Then, we found a proper coloring of that graph. Now we're at the home stretch! Consider any point inside the original polygon. By construction of the triangulation, that point must lie in one of the triangles. The point is thus visible, in particular, from all three corners of that triangle. In other words, there is a blue-colored corner that sees the point, as well as a red-colored corner and a yellow-colored corner – contemplate Figure 6 if you want a visual reminder. Hence, if we place guards at all corners with the same color, then the whole gallery is guarded. Moreover, since all corners are given precisely one color we have that

number of blue corners + number of red corners + number of yellow corners = number of corners in total.

So far, we have written no equations at all (shocking, isn't it!), but this one we might just have to rewrite as an equation. If n is the total number of corners and n_{blue}, n_{red} and n_{yellow} are the number of blue, red respectively yellow corners then what we just wrote is the same as

$$n_{\text{blue}} + n_{\text{red}} + n_{\text{yellow}} = n.$$

Let's mention one more prominent proof-method of mathematicians: contradiction. To reach a contradiction, assume that all three values n_{blue}, n_{red} and n_{yellow} are strictly larger than $n/3$. Then, if we sum the three values together we must get something that is strictly larger than $n/3+n/3+n/3 = n$. So on the one hand we have

$$n_{\text{blue}} + n_{\text{red}} + n_{\text{yellow}} > n,$$

and on the other

$$n_{\text{blue}} + n_{\text{red}} + n_{\text{yellow}} = n.$$

But that means that we have proved that n is strictly larger than itself, and that's absurd! There exist no such number. This means that something must be wrong, and that something is the original assumption that all three values n_{blue}, n_{red} and n_{yellow} are strictly larger than $n/3$. Therefore, at least one of n_{blue}, n_{red} and n_{yellow} is equal to or smaller than $n/3$, and we can place guards at the corners with the corresponding color. And to reiterate: placing guards there, means the whole polygon is guarded. That's the proof!

Is there anything more to say? Well, perhaps not. But look at all we have covered! We went from a quite tangible, real-world problem to a geometric problem stated in terms of polygons and lines therein, and then made that polygon into a graph, rephrasing and throwing away redundant information. We colored the vertices of a graph in a certain way, and used those colors to finally place our guards. The specific requirements on the coloring made sure both that the placements of the guards made the whole thing guarded, and that the number of guards was forced to be at most a third of the number of corners. With all this, you've glimpsed into the world of *real* mathematics; it is quite different from trigonometric identities, calculations with percentages and all that stuff you did in school.

References and Further Reading

A little historic note may be in place. In the world of mathematics The Art Gallery Problem is a relatively new one, posed in a letter by the mathematician Victor Klee in 1973 and almost immediately proved by the recipient Václav Chvatál. The original proof of Chvatál was not lengthy – just a couple of pages – but then Steve Fisk shortened it to a single page in 1978. A single page! In fact, the proof we have outlined in this text is that of Fisk. It is quite celebrated by mathematicians as a *beautiful* proof, evident by its appearance in a book called *Proofs from THE BOOK* which collects proofs of theorems generally acclaimed to have an aesthetically pleasing nature. There is even a full book whose content is completely devoted to The Art Gallery Problem and related material. I recommend both books, which can be read by anyone with mathematical interest and a little bit of stubbornness.

Aigner, M. and G. Ziegler (2018). *Proofs from THE BOOK*. 6th ed. Springer Berlin.

Chvátal, V. (1975). "A combinatorial theorem in plane geometry". In: *Journal of Combinatorial Theory, Series B* 18.

Fisk, S. (1978). "A short proof of Chvátal's Watchman Theorem". In: *Journal of Combinatorial Theory, Series B* 24.

O'Rourke, J. (1987). *Art gallery theorems and algorithms*. Oxford University Press, Inc.