Math 7h — Professor: Padraic Bartlett

Lecture 4: Trees and Art Galleries

Week 2 — *UCSB 2014*

1 Prelude: Graph Theory

In this talk, we're going to refer frequently to things called **graphs** and **trees**. If you haven't seen these before, we define these concepts here:

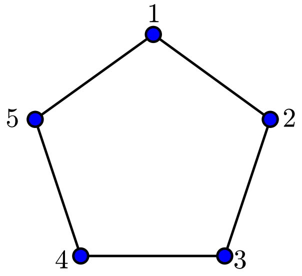
Definition. A
textbfgraph G with n vertices and m edges consists of the following two objects:

1. a set $V = \{v_1, \ldots v_n\}$, the members of which we call G's **vertices**, and
2. a set $E = \{e_1, \ldots e_m\}$, the members of which we call G's **edges**, where each edge e_i is an unordered pair of distinct elements in V, and no unordered pair is repeated. For a given edge $e = \{v, w\}$, we will often refer to the two vertices v, w contained by e as its endpoints.

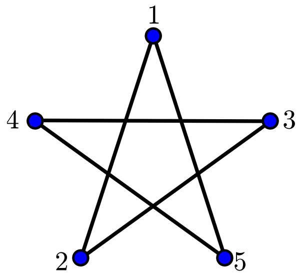
Example. The following pair (V, E) defines a graph G on five vertices and five edges:

- $V = \{1, 2, 3, 4, 5\}$,
- $E = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{5, 1\}\}$.

Something mathematicians like to do to quickly represent graphs is **draw** them, which we can do by taking each vertex and assigining it a point in the plane, and taking each edge and drawing a curve between the two vertices represented by that edge. For example, one way to draw our graph G is the following:



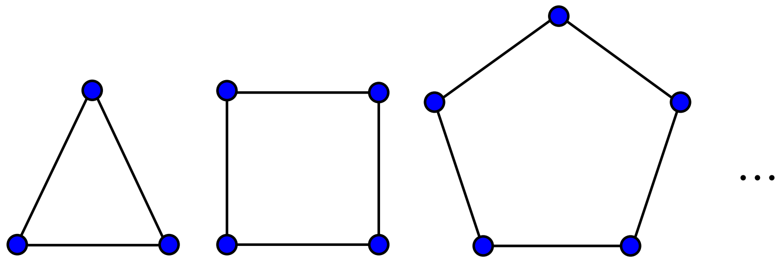
However, this is not the only way to draw our graph! Another equally valid drawing is presented here:



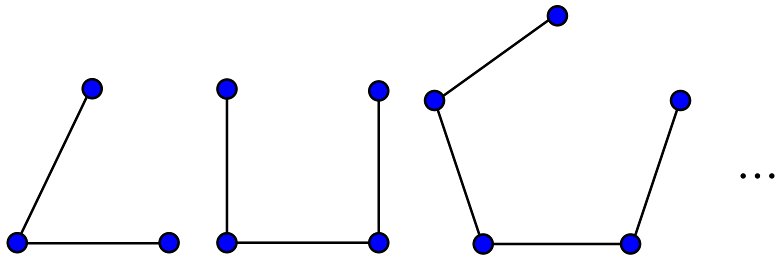
In general, all we care about for our graphs is their vertices and their edges; we don't usually care about how they are drawn, so long as they consist of the same vertices connected via the same edges. Also, we usually will not care about how we "label" the vertices of a graph: i.e. we will usually skip the labelings on our graphs, and just draw them as vertices connected by edges.

Some graphs get special names:

Definition. The **cycle graph** on n vertices, C_n, is the graph on the vertex set $\{v_1, v_2, \dots v_n\}$ with edge set $E(C_n) = \{\{v_1, v_2\}, \{v_2, v_3\}, \dots \{v_{n-1}, v_n\}, \{v_n, v_1\}\}$. The cycle graphs C_n can be drawn as n-gons, as depicted below:



Definition. The **path graph** on n vertices, P_n, is the graph on the vertex set $\{v_1, v_2, \dots v_n\}$ with edge set $E(C_n) = \{\{v_1, v_2\}, \{v_2, v_3\}, \dots \{v_{n-1}, v_n\}\}$. The path graphs P_n can be drawn as paths of length n, as depicted below:

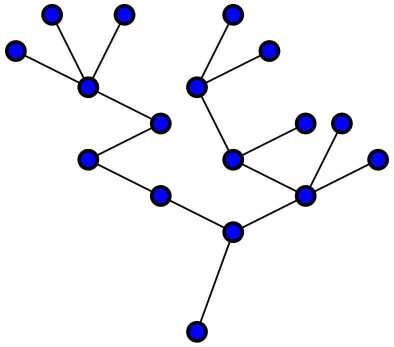


Definition. Given a graph G and another graph H, we say that H is a **subgraph** of G if and only if $V(H) \subset V(G)$ and $E(H) \subset E(G)$.

Definition. Given a graph G, we call G **connected** if for any two vertices $x, y \in V(G)$, there is a path that starts at x and ends at y in our graph G.

Definition. If a graph G has no subgraphs that are cycle graphs, we call G **acyclic**. A **tree** T is a graph that's both connected and acyclic. In a tree, a **leaf** is a vertex whose degree is 1.

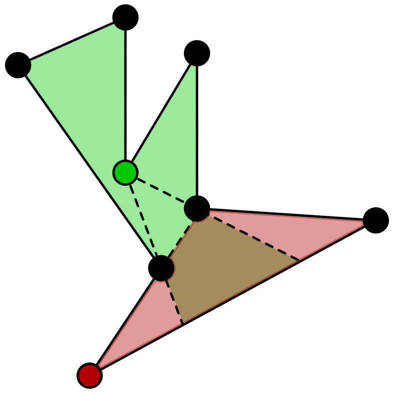
Example. The following graph is a tree:



2 The Art Gallery Problem

Consider the following question:

Question 1. *Suppose that you have an art gallery that is shaped like some sort of n-polygon, and you want to place cameras with $360°$-viewing angles along the vertices of your polygon in such a way that the entire gallery is under surveillance. How many cameras do you need?*



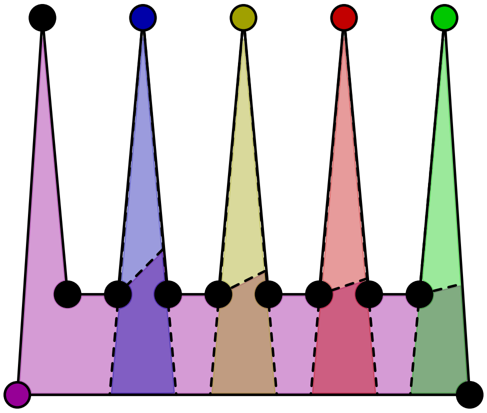
A gallery guarded by 2 guards, Red and Green.

One trivial upper bound you can come up with is n guards: just put one guard on each vertex of our polygon with n sides!

Can we do better? As it turns out, we can!

Claim. (Chvátal) You need at most $\lfloor n/3 \rfloor$-many cameras to guard a n-polygon.

It bears noting that this bound of $\lfloor n/3 \rfloor$ is sharp. Consider the following art gallery:



A crown-shaped art gallery.

In this sort-of "crown-shaped" art gallery, each prong of the crown (i.e. triangle) needs to have a guard on one of its three vertices to guard the entire triangle, as no other vertices can "see" the entirety of that prong. Therefore, you need one guard for each prong; i.e. $n/3$ guards, for a crown with $n/3$ prongs (i.e. n vertices.)

To prove Chvátal's theorem, we need a few lemmas first:

Lemma 2. *If G is a n-polygon with $n \geq 4$, then there is some line segment formed by two of the vertices in G that lies entirely in G.*

Proof. Let v be the leftmost vertex of G. (If there is a tie, take v to be the top leftmost vertex of G.) Let u and w be v's neighbors, and examine the line segment $\overline{uw}$. If this lies entirely in G, great! Otherwise, it must cross some edge of G; consequently, there must be a vertex of G that lies inside of the triangle spanned by the three points u, v, w. Let x be the vertex furthest from the line segment $\overline{uw}$ that lies in this triangle. Then, look at the line segment $\overline{vx}$; because x is the furthest point in Δuvw from $\overline{uw}$, there can't be any edges of G that are crossed by this line segment (as one of their endpoints would necessarily be closer to v.) So $\overline{vx}$ lies entirely in G. □

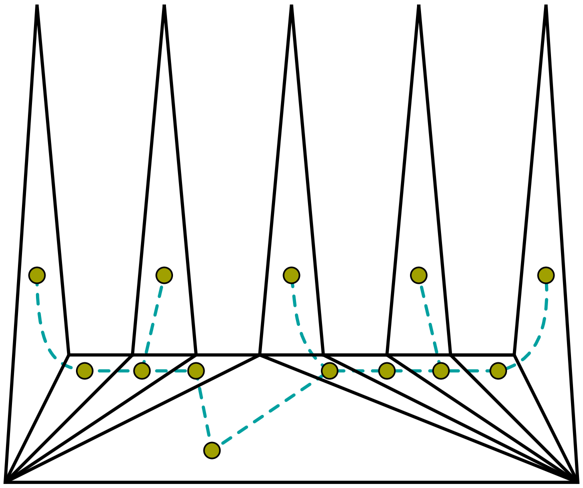
Corollary 3. *Any n-polygon can be divided into $n-2$-triangles.*

Proof. Using the process above, repeatedly divide our n-polygon into a pair of smaller polygons, one with k vertices and the other with $n+2-k$ vertices, until all of these polygons are triangles. By induction, it is not hard to see that the number of these triangles is $n-2$. □

So: we can turn any polygon into a number of triangles stuck to each other! We use this to turn any art gallery on n vertices into a **graph** on $n-2$ vertices, as follows:

- Start by taking our polygon G and turning it into a collection $\{T_i\}_{i=1}^{n-2}$ of triangles.
- For each triangle T_i, associate a vertex t_i.
- Connect two vertices t_i, t_j with an edge if their corresponding triangles T_i, T_j share a face.

Call this graph T' the **dual graph** of T.



Turning the crown into a tree.

This is a graph! Furthermore, it's a fairly special kind of graph: it's a tree! We prove this here:

Lemma 4. *Let G be a polygon, T be a triangulation of G performed as above, and let T' be the dual graph to this triangulation (i.e. put a vertex in the center of every face of T , and connect two faces iff they share an edge.) This graph is a tree.*

Proof. Let T be our triangulated polygon. In our construction above, each of the edges of T' corresponds to a diagonal of the polygon G, that divides our polygon into two distinct smaller polygons. Because cutting our polygon G along one of those diagonals will always divide the polygon into two disconnected pieces, doing so will always result in two triangles that are no longer connected by a chain of triangles with adjacent faces!

In other words: in the dual graph T' that we made above, removing any edge disconnects our graph! So our graph has the following properties:

- There is a path between any two vertices because our polygon is connected.
- There is at most one path between any two vertices, because if there were two paths and we cut an edge on one path, our graph would not be disconnected (we could simply use the other path to connect vertices!) In other words, our graph is **acyclic**, because a cycle exists in our graph if and only if there are two different paths connecting two vertices! (Prove this to yourself if you are skeptical.)

Therefore T is connected and acyclic: i.e. it's a tree! □

This tree is remarkably useful; in particular, we can use its structure to create a system for assigning guards! We do this here:

Lemma 5. *Take a polygon G that has been triangulated as described earlier. Then we can color each of the vertices of G either red, blue or green, so that each triangle contains one vertex of each color.*

Proof. For our triangulated polygon G, take the dual graph/tree T' that we constructed above, and pick some vertex t_0 in it. For all relevant integers k, let T'_k be the collection of vertices that are distance k away from v, for every k. (The **distance** of two vertices from each other is the length of the shortest path between them.)

Color the vertices of T as follows:

- Take the triangle in G associated to t_0 and color its three vertices red, green and blue.
- Suppose we've colored all of the vertices attached to triangles with corresponding vertices in T'_i, for some i. Now, look at the triangles corresponding to vertices in T'_{i+1}. Each triangle associated to a vertex t_{i+1} in this set shares exactly one edge with some triangle associated to a vertex in T'_i; this is because if our vertex is distance $i+1$ from t_0, then (by taking the path of distance $i+1$ and walking one step closer to t_0) there is an adjacent vertex (and thus face-sharing triangle) at distance i, i.e. in T'_i.

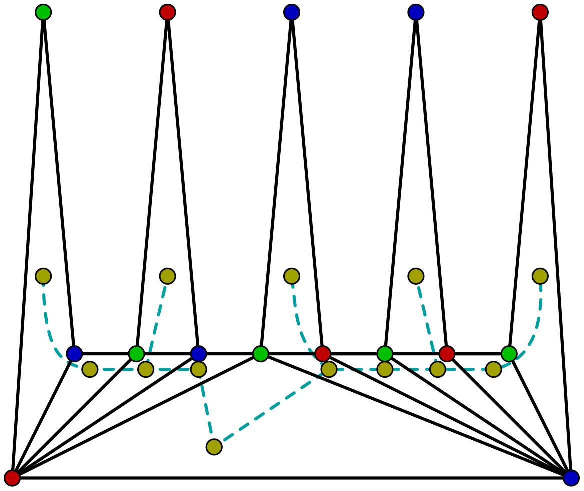
  Furthermore, because T' is a tree, there is exactly one edge from any t_{i+1} to vertices in the set $\bigcup_{j=0}^{i+1} T'_j$. This is because the existence of any other edge would create a cycle, because it would give us two distinct paths to t_0!

  Therefore, the triangle associated to t_{i+1} shares a face with **only one other** triangle in all of the sets that we've already colored! Therefore, only two of its vertices have been assigned colors. Thus, there is always some spare third color to use to color its third vertex! Use this to color its third vertex, and repeat for all vertices in T_{i+1}.
- Repeat until every vertex in T is colored. Note that each triangle has one vertex of each color.

□

Corollary 6. *You need at most $\lfloor n/3 \rfloor$-many cameras to guard a n-polygon.*

Proof. By the above, create a triangulation and 3-coloring of our polygon G with the colors $\{R, G, B\}$. Now, station guards at whichever color is used the least number of times in this triangulation! Each guard can see everything in their assigned triangle by construction. Therefore, the entire art gallery is guarded. □



To guard this crown, simply pick one of (red, green, blue,) and station guards at vertices of that color.