NUMB3RS Activity: No Place Left to Hide
Episode: The Art of Reckoning

Topic: Pursuit evasion, families of graphs, and proof **Grade Level:** 9 - 12
Objective: Using vertex-edge graphs as models, determine the minimum number of searchers needed to find and trap a fugitive
Time: 20 - 30 minutes

Introduction

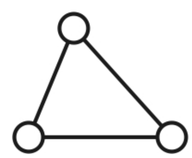
In "The Art of Reckoning," Don suspects that a detailed confession he receives from a killer may be a trick to allow an escape from the Secure Housing Unit at the prison. Along with a discussion of vulnerability analysis, Charlie uses the example of a spy trying to get through enemy lines. He says that he can "calculate a ratio of vulnerable points versus the resources required to defend each point, to determine probabilities of penetration." Charlie's approach is best used with limited resources, placing them where they will be the most effective. A related but simpler method called "pursuit evasion" can be used to determine exactly what resources are necessary to guarantee that enemy lines are impenetrable. In this activity, students will use vertex-edge graphs to find out how many searchers it will take to find and capture a fugitive.

Discuss with Students

In the episode, the audience is shown the schematic of a fort under siege, with a spy trying to get through enemy lines. In this activity, the schematics are modeled by vertex-edge graphs, in which the edges represent corridors or tunnels, and the vertices represent intersections or rooms.

Pursuit evasion is generally characterized by the question "How many searchers does it take to find and trap a fugitive who can move infinitely fast?" The minimum number of searchers is called the *search number* of a graph G, designated by $s(G)$. This activity is an exploration that encourages students to find patterns in the properties of graphs that affect the search number. Make it clear to students that a searcher cannot "see" along the edges, and must search an edge by traveling it.

Consider the example shown at right. A single searcher cannot catch the fugitive, but will just endlessly chase him around the triangle. This graph requires two searchers, $s(G) = 2$.



The first six graphs on the student page are samples of different families of graphs, named and described in the answers below. These answers also include an algorithm that can form the backbone of a more formal general proof for each type of graph. Discussion of the solutions and their generalizations can provide students with experience in elementary graph theory, problem solving, and proof. For each of the graphs in this activity, encourage students to describe how the fugitive can elude one less searcher than the number given.

Student Page Answers:

1. *$s(G) = 2$. This graph is a cycle with six vertices, notated as C_6. The example in "Discuss with Students" above is a C_3. The fugitive can elude a single searcher, but if a second one even just stands still on a vertex, the other searcher will force the capture.* **2.** *$s(G) = 4$. This is a simpler version of graph #4, called a wheel. All four searchers start on one of the vertices that have three edges. Two searchers move to the "hub" while*

L. Charles (Chuck) Biehl, The Charter School of Wilmington, Wilmington, DE

the other two split up to search the edges leading to the next vertices with three edges. The searchers in the hub each search an edge to where the other searchers are waiting, and then the pairs search the last edges to the final vertex, where all four searchers meet up again. **3.** *s(G) = 3. This graph is a fan with seven vertices, notated as* F_7*. A fan is a set of vertices connected in a line (called a path), with all of the vertices in the path connected to a single vertex, called the hub. All three searchers start on the hub. One searcher searches the edge to the vertex at the left end of the path, and then searches to the next vertex on the path, being met by one of the searchers from the hub who searched the edge leading from the hub. The two proceed together to the next vertex on the path, where one searches the edge back to the hub and the other searches the edge to the next vertex on the path. They repeat this process until they come to the end of the path.* **4.** *s(G) = 4. A wheel is a cycle graph with all vertices in the cycle connected to a hub vertex. Because this graph has seven vertices, it is notated as* W_7*. All four searchers start in the hub. Two stay there while the other two together search a single edge to the cycle. These two then split up and search one edge each. One of the searchers from the hub searches each edge to where the other two searchers are, then returns to the hub. The two searchers on the cycle each search another edge, followed by the searcher from the hub again, etc., until the two searchers on the cycle meet up.* **5.** *s(G) = 3. This graph is a tree, defined as a graph that is connected (all one piece) but contains no circuits. Because of their nature, it is possible to construct trees that require arbitrarily many searchers. Vertices that have only one edge are called "leaves"; all other vertices are called "internal" vertices. In this example, the searchers start at the internal vertex connected to the other three internal vertices. One stays behind and the other two search the edge to another internal vertex. Here they split up and search each edge to the leaves. They return together to the searcher who stayed behind, and then repeat the process on the other "branches" of the tree.* **6.** *s(G) = 6. This is a complete graph with six vertices, one in which every possible pair of vertices is directly connected by an edge. It is notated as* K_6 *(recall that "C" is for cycle, which is why "K" is used here.) Start with all of the searchers on a single vertex. Send five of them to search each of the five edges leading from this vertex. This process "clears" the starting vertex. The last searcher on the starting vertex can now systematically search all of the remaining edges. To generalize,* K_1 *is trivial,* K_2 *requires one, and* K_3 *requires two. The sample in "Discuss with Students" above, in addition to being a* C_3*, is also a* K_3*. For* $n \geq 4$*,* K_n *requires n searchers, using the same method as described for* K_4*.* **7.** *s(G) = 4. This is the graph used for the "Extensions" and is not any special type of graph. Start with all four searchers in the top left vertex. One searches one edge down, one searches one edge right, and the other two search to the center of the graph. The "extra" searcher in the center searches the edges to where the other two searchers are waiting (and back). The "outside" searchers then search two more edges each and then wait. The "extra" searcher in the center repeats the search process for the two edges where the single searchers are waiting (and back). Finally, all four searchers search the last edge available to them, and they meet in the bottom right vertex.*

L. Charles (Chuck) Biehl, The Charter School of Wilmington, Wilmington, DE

Name: ___________________________________ Date: ________________

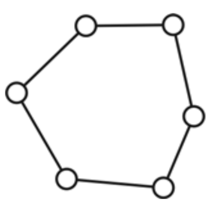
NUMB3RS Activity: No Place Left to Hide

In “The Art of Reckoning,” Don suspects that a detailed confession he receives from a killer may be a trick to allow an escape from the Secure Housing Unit at the prison. Along with a discussion of vulnerability analysis, Charlie uses the example of a spy trying to get through enemy lines. He says that he can “calculate a ratio of vulnerable points versus the resources required to defend each point, to determine probabilities of penetration.” Charlie’s approach is best used with limited resources, placing them where they will be the most effective. A related but simpler method called “pursuit evasion” can be used to determine exactly what resources are necessary to guarantee that a spy cannot penetrate enemy lines. In this activity, you will use vertex-edge graphs to find out how many searchers it will take to find and capture a fugitive.

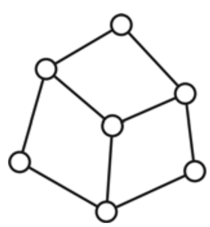
This activity uses vertex-edge graphs as models for floor plans where edges represent tunnels or corridors and vertices are intersections or rooms. Pursuit evasion asks the question “How many searchers does it take to find and trap a fugitive in the graph?” The answer is called the *search number* of the graph. The fugitive can move infinitely fast, and can hide on an edge between two vertices. The searchers cannot see along the edges, so they must search by traveling every edge. Searchers may start on any vertex, and multiple searchers can share a vertex or edge.

For each graph, determine the search number, $s(G)$, which is the minimum number of searchers required to guarantee that the fugitive will be caught. Look for patterns in how the properties of the graphs change. Try to prove your results for different “families” of graphs that share the same properties but have different numbers of vertices.

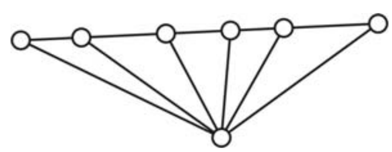
1.



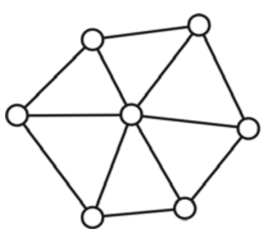
2.



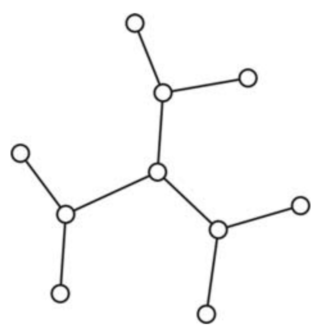
3.



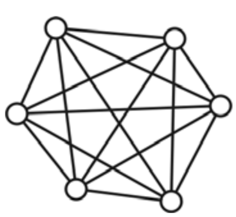
4.



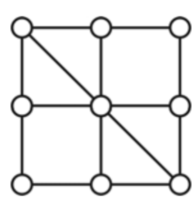
5.



6.



7.



L. Charles (Chuck) Biehl, The Charter School of Wilmington, Wilmington, DE

The goal of this activity is to give your students a short and simple snapshot into a very extensive mathematical topic. TI and NCTM encourage you and your students to learn more about this topic using the extensions provided below and through your own independent research.

Extensions

Introduction

This activity has provided the essential concepts behind the mathematics of pursuit evasion. In a time when the concern for security has reached unprecedented levels, research on efficient methods of achieving a totally secure environment becomes increasingly important. When extended from graphs to actual floor plans or schematics, pursuit evasion has many applications, such as searching a building following a disaster, clearing an area of obstacles, or searching for an escapee.

For the Student

This activity required that an edge be traveled in order to be searched. A variation of the search number is called the “node search number,” where the searcher can “see” along an edge to the next vertex. Determine the node search number of each graph from this activity. Think about situations in which each type of search number is more appropriate.

A two-player game called “The Hunter and the Rabbit” models pursuit evasion on a graph. The players each place a marker on a vertex. For each turn, the hunter moves *randomly* to any adjacent vertex, and simultaneously the rabbit may do the same or not move at all, also randomly. If the rabbit is exactly one edge away from the hunter, it can “see” the hunter and will not choose that edge. The object for the hunter is to land on the same vertex as the rabbit in the fewest number of turns. The rabbit’s goal is to survive the greatest number of turns. Graph #7 in this activity was designed by Heidi Van Every as a game board for this game. Try the game on this and different graphs, or even change the rules. Possible rule changes could be that the rabbit has the power to “jump” to any vertex in the graph, like through secret tunnels.

Related Topic

A major goal in pursuit evasion is to use robot searchers. Research topics include robot sensory technology, automatic communication between robots, and efficient algorithms for search patterns for teams of robots, including self-organization of search strategies. Various types of robot searching are actually extensions of the “Art Gallery” problem. This problem asks how many *stationary* cameras, searchers, robots, or guards are necessary to keep an area, usually constructed as a polygon, under simultaneous surveillance.

Additional Resources

For some animations of robot searches, see Brian Gerkey’s site at Stanford:
http://ai.stanford.edu/~gerkey/research/pe

The Embedded Network Laboratory (ENL) at USC has done research on pursuit-evasion games using robots and wireless networks: **http://enl.usc.edu/projects/peg/index.html**

For a *NUMB3RS* activity on the art gallery problem, go to
http://education.ti.com/exchange and search for “5984.”

L. Charles (Chuck) Biehl, The Charter School of Wilmington, Wilmington, DE