Activity 9

The Muddy City—*Minimal Spanning Trees*

Summary

Our society is linked by many networks: telephone networks, utility supply networks, computer networks, and road networks. For a particular network there is usually some choice about where the roads, cables, or radio links can be placed. We need to find ways of efficiently linking objects in a network.

Curriculum Links

- ✓ Mathematics: Geometry Level 2/3 and up. Exploring shape and space: Finding the shortest paths around a map

Ages

- ✓ 9 and up

Skills

- ✓ Problem solving

Materials

Each child will need:

- ✓ Workshop Activity: The muddy city problem (page 78)
- ✓ Counters or squares of cardboard (approximately 40 per child)

The materials seem solid since they are all physical things (except for any instructions which should be made into a digital copy for screen readers)

The Muddy City

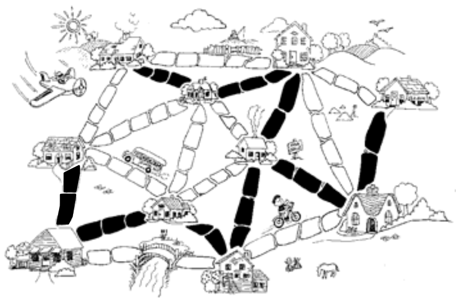
Introduction

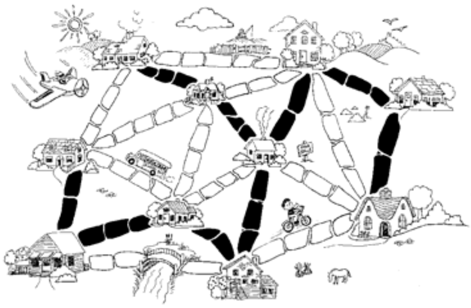
This activity will show you how computers are used to find the best solutions for real-life problems such as how to link power lines between houses. Have the children use the worksheet on page 78, which explains the 'Muddy City' problem.

Follow-up discussion

Share the solutions the children have found. What strategies did they use?

One good strategy to find the best solution is to start with an empty map, and gradually add counters until all of the houses are linked, adding the paths in increasing order of length, but not linking houses that are already linked. Different solutions are found if you change the order in which paths of the same length are added. Two possible solutions are shown below.





Another strategy is to start with all of the paths paved, and then remove paths you don't need. This takes much more effort, however.

Where would you find networks in real life?

Computer scientists call the representations of these networks "graphs". Real networks can be represented by a graph to solve problems such as designing the best network of roads between local cities, or aeroplane flights around the country.

There are also many other algorithms that can be applied to graphs, such as finding the shortest distance between two points, or the shortest route that visits all the points.

Worksheet Activity: The Muddy City Problem

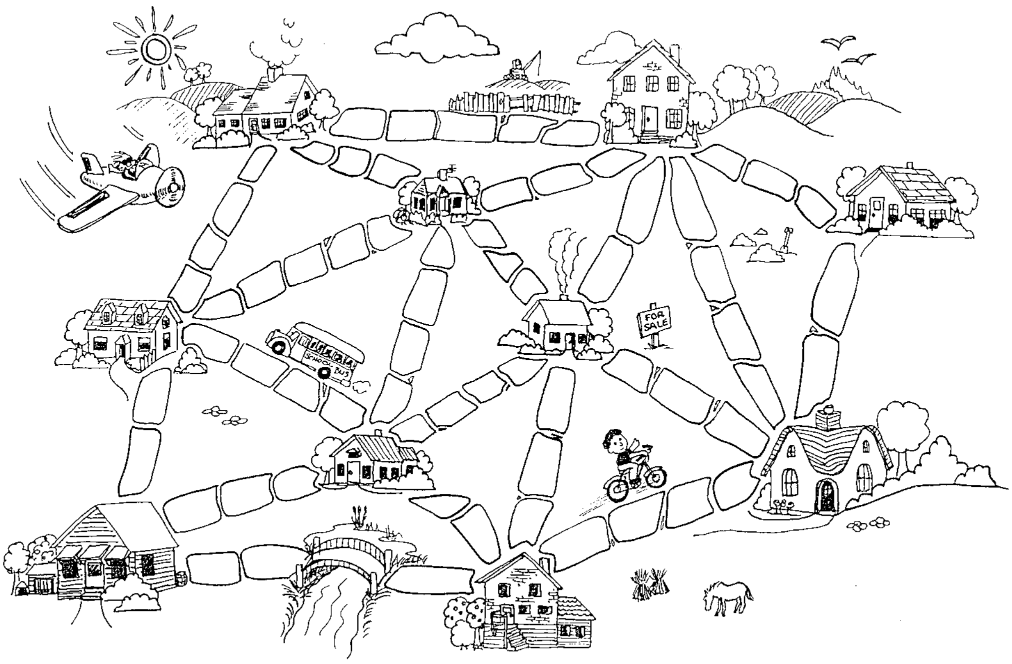
Once upon a time there was a city that had no roads. Getting around the city was particularly difficult after rainstorms because the ground became very muddy—cars got stuck in the mud and people got their boots dirty. The mayor of the city decided that some of the streets must be paved, but didn't want to spend more money than necessary because the city also wanted to build a swimming pool. The mayor therefore specified two conditions:

1. Enough streets must be paved so that it is possible for everyone to travel from their house to anyone else's house only along paved roads, and

2. The paving should cost as little as possible.

Here is the layout of the city. The number of paving stones between each house represents the cost of paving that route. Find the best route that connects all the houses, but uses as few counters (paving stones) as possible.

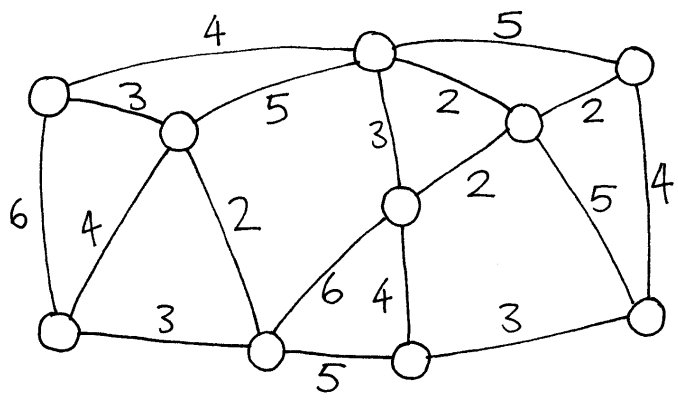
What strategies did you use to solve the problem?



Here is the big hurdle. How do we represent graphs? How do we represent the following graph theory? We'll need some way to either have a raised picture or some sort of model in the middle of the classroom that the students can use. I can't think of any efficient textual ways to represent the data (which is express graphically) on the next page. All we know is that this needs to be reworked for certain.

Variations and extensions

Here is another way of representing the cities and roads:



The houses are represented by circles, the muddy roads by lines, and the length of a road is given by the number beside the line.

Computer scientists and mathematicians often use this sort of diagram to represent these problems. They call it a *graph*. This may be confusing at first because "graph" is sometimes used in statistics to mean a chart displaying numerical data, such as a bar graph, but the graphs that computer scientists use are not related to these. The lengths do not have to be drawn to scale.

Make up some of your own muddy city problems and try them out on your friends.

Can you find out a rule to describe how many roads or connections are needed for a best solution? Does it depend on how many houses there are in the city?

Again, how do we represent graph theory? How do we have the kids make their own muddy city problems? This definitely needs to be reworked. The questions, however, can stay if we can find a way to work out this activity.

What's it all about?

Suppose you are designing how a utility such as electricity, gas, or water should be delivered to a new community. A network of wires or pipes is needed to connect all the houses to the utility company. Every house needs to be connected into the network at some point, but the route taken by the utility to get to the house doesn't really matter, just so long as a route exists.

The task of designing a network with a minimal total length is called the *minimal spanning tree* problem.

Minimal spanning trees aren't only useful in gas and power networks; they also help us solve problems in computer networks, telephone networks, oil pipelines, and airline routes. However, when deciding the best routes for people to travel, you do have to take into account how convenient the trip will be for the traveller as well as how much it will cost. No-one wants to spend hours in an aeroplane taking the long way round to a new country just because it is cheaper. The muddy city algorithm may not be much use for these networks, because it simply minimizes the *total* length of the roads or flight paths.

Minimal spanning trees are also useful as one of the steps for solving other problems on graphs, such as the "travelling salesperson problem" which tries to find the shortest route that visits every point in the network.

There are efficient algorithms (methods) for solving minimal spanning tree problems. A simple method that gives an optimal solution is to start with no connections, and add them in increasing order of size, only adding connections that join up part of the network that wasn't previously connected. This is called Kruskal's algorithm after J.B. Kruskal, who published it in 1956.

For many problems on graphs, including the "travelling salesperson problem", computer scientists are yet to find fast enough methods that find the best possible solution.

Solutions and hints

Variations and extensions (page 79)

How many roads or connections are needed if there are n houses in the city? It turns out that an optimal solution will always have exactly n–1 connections in it, as this is always sufficient to link up the n houses, and adding one more would create unnecessary alternative routes between houses.