Many Problems, One Solution

Teacher Notes to support Tenderfoot Unit 2: Clever Stuff For Common Problems – Going beyond simple algorithms

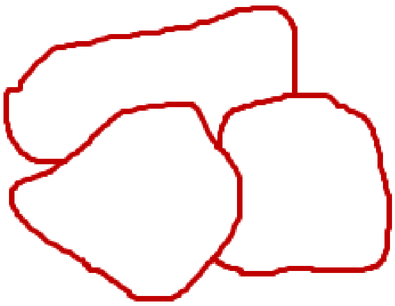
Three activities that introduce 'map colouring' problems, abstract representation as graphs, the algorithm to identify the minimum colours required to colour a map and its application to a real world problem.

Preparation required:
1 Quibble, Crouch Positions Sheet
Set of pavement chalks or lots of string for first activity.

Quibble, Crouch, Cornerstone and Quoink

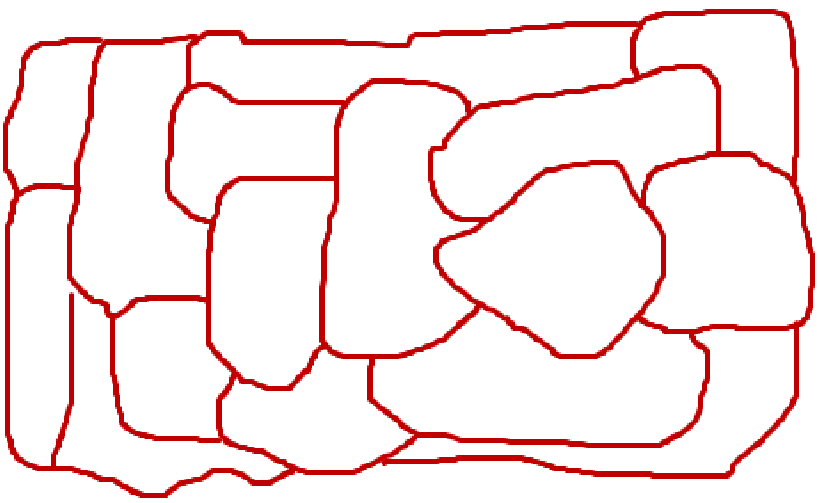
This activity is weather dependent, although it could be done in an open indoor space. It will require some pavement chalks (or string) and probably works best in groups of approximately 12 to 17. The idea was taken from the now defunct MegaMath project. MegaMath was a project of the Computer Research and Applications Group at Los Alamos National Laboratory.

Each child in turn draws a small area in the playground to stand in. An example of the first three areas of a 'map' are shown right.

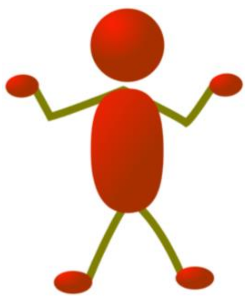


Each area should be connected to at least one of the existing areas. Encourage children to draw shapes that connect to more than one other area if possible (without getting too ridiculous!).

Gradually build up the map, with each child adding and then standing in their area until everyone is standing on the map.



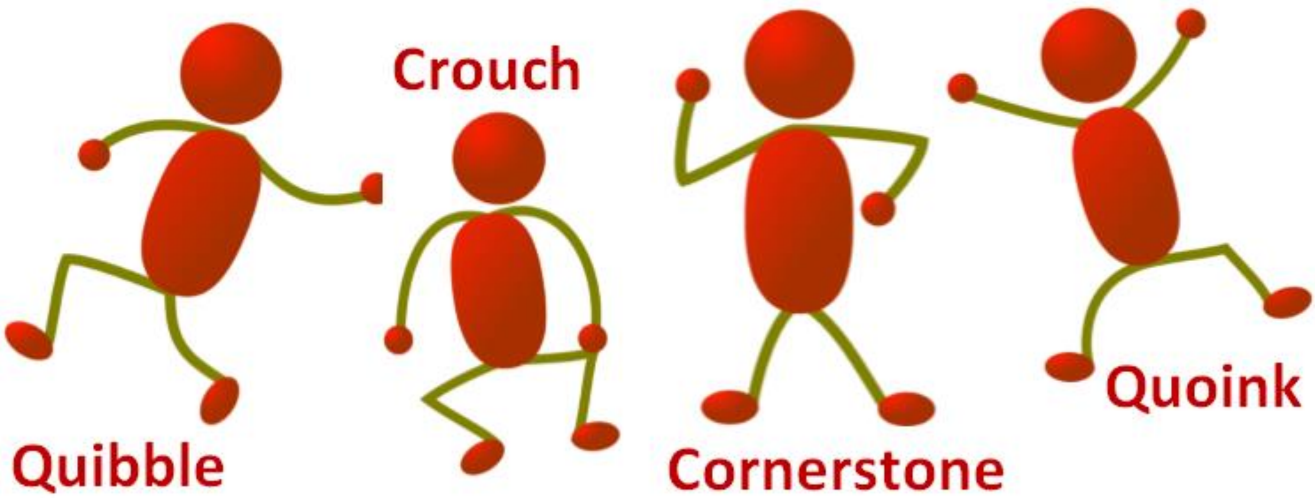
If you have any particularly boisterous children, you may wish to retain a couple initially to help you direct operations and check everyone is doing it right. Once the activity starts, you can always add them in at a later stage by adding extra spaces to the map. This can be a useful extension, particularly if they solve the initial challenge quickly.



Each child will start in a confused state. Model the stance shown left. This indicates they don't know what to do.

The challenge is for the students to arrange themselves into one of four positions, Quibble, Crouch, Cornerstone and Quoink.

Each stance is explained overleaf.



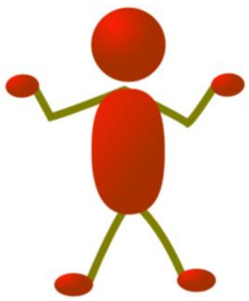
The challenge is for the students to arrange themselves into one of the four positions:

Quibble: standing on one leg (they can swap legs if they are tired!)

Crouch: squatting,

Cornerstone: standing straight and tall, hand on hip and clenched first salute and

Quoink: jiggling about and waving their arms above their heads.



The only rule is that they must not adopt the same position as any of their neighbours that their shape connects with.

If they don’t know what to do, or can’t find a position to satisfy the rule, they stay in a confused state.

The challenge, as a group is to see how long it takes until no-one is still confused.

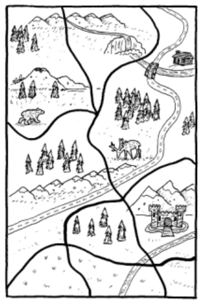
If silly positions are potentially too disruptive, each child can be issued with cards bearing the position names which they simply hold aloft.

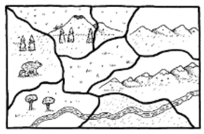
If the group find a solution quickly, remove one of the positions and see if they can find a solution with just three different positions.

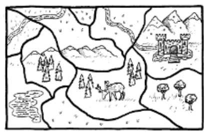
A final extension might be to create a map by drawing closed loops, overlaid on each other. Challenge the group to find the smallest number of positions needed to satisfy the rule that no two adjoining areas adopt the same stance. The discrete areas of a map created this way can always be filled with just two positions.

Back in the classroom you can discuss the success (or otherwise) of their efforts. If you have a camera, a snapshot of the ‘map’ might be useful to display on the board.

Preparation required:
3 maps for each student, spare plain paper.
Several sets of coloured pencils







The Poor Cartographer's Problem

Three simple maps are included to introduce map colouring and the 'poor cartographer's problem'. Use the larger map first – possibly as a group activity. The challenge for students is to find the minimum number of colours required to colour each country. Countries that share borders should not be the same colour. We want children to discover the 'has to be' rule, and identify ways to check existing colours before adding another. In this case, the map can be coloured using just two colours. We can also use it to clarify the point about countries meeting at a point, which does not constitute a shared border.

The two smaller maps requires three and four colours respectively. Set these as independent challenges. An extension task might be to ask students to design their own map that requires five colours to complete. Whilst students may think they have a designed such a map, all maps drawn on a flat surface require only four colours.

It can be pointed out that the conjecture that any map can be coloured using only four colours was formulated in 1852, but it was not proved until 1976. Computer science is full of unsolved problems, and knowing that the four-colour theorem was proved after more than 120 years of research is encouragement for those Computer Scientists looking for solutions to hard problems today. Another extension challenge is to draw a map consisting of overlaid, closed loops. What is the minimum number of colours needed for this? Any map constructed in this fashion only needs two colours.

As a class you can consider some of the problems we might face tackling a harder problem, such as the map of Europe. What strategies might we adopt? Would we start with any specific country e.g. the one with the most borders? How do we know which country has the most borders anyway in a large map? We need to abstract away the unnecessary detail in a map, then develop an algorithm to solve it. The associated slides demonstrate how to develop a graph where nodes represent countries and edges their shared borders.

To apply the algorithm we can represent the graph as an 'adjacency list'. This is a list of nodes, each with an attached list of connections. The slides walk through implementing the algorithm. This is a two stage process: Stage 1 determines the best order to colour the nodes and Stage 2 assigns a legal colour. The slides allow you to work through this as a class, pausing for questions, each step initiated 'on click'.

For Stage 1 the node with the smallest number of connections is selected first. Take care to point out how, when a node is selected, it is removed from the adjacency lists of any nodes remaining. This is the key stumbling block to implementing the algorithm. Where several nodes have the smallest number of connections, any can be selected. Repeatedly applying this leads to a new list of ordered nodes being created.

Stage 2 involves selecting nodes in reverse order from the new list. There is the potential here to investigate how we might reverse the order of a list. Had we built the list as a stack, the first node selected in stage 1 would have been at the bottom and the last at the top. Taking items from the top of the stack selects them in the reverse order to that when added. The last one in, is the first one out. For each node selected, its connections are listed, and a colour assigned. We will always assign a colour used already if we can. However, we cannot select a colour assigned to another node, if that node is listed as connected.

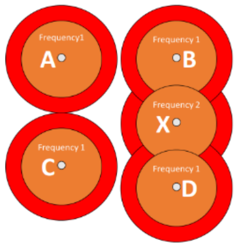
Preparation required:
Mobile Phone Mast Problem activity sheet for each student.

Mobile Phone Mast Problem

This activity illustrates how an algorithm for one problem (map colouring) can be applied to a seemingly unrelated one. It provides a real world context, reinforces the idea of abstract representation of the problem as a graph, and it encourages children to have a go at implementing the map colouring algorithm independently.

Mobile phone use has exploded over the last twenty years. To satisfy demand, network operators have built ever increasing numbers of base station transmitters / receivers. The problem they face will be familiar to anyone who has a wireless doorbell. Where many devices use wireless communication, to avoid interfering with each other, they must use different frequencies. Small devices work in the unregulated frequency ranges. They usually avoid interference by having several frequencies they can potentially operate on.

Telecommunications is a more serious matter. Telecoms operators must operate within a regulated range. Each operator must purchase the right to transmit on a particular frequency. Each base station has a transmission range as shown on the schematic. Base stations transmitting on the same frequency cannot overlap without causing interference. To cover the gap, an extra station (X), working on a different frequency is required. Each telecoms provider will want to maximise mobile phone reception, but minimise the number of frequencies it has to purchase.



	1	2	3	4	5	6
1	-	35	245	12	400	112
2	35	-	175	47	149	233
3	245	175	-	111	365	411
4	12	47	111	-	65	350
5	400	149	365	65	-	211
6	112	233	411	350	211	-

The table shows six base stations owned by a hypothetical telecoms provider. The figures indicate the distance between each mast. Masts that are within 150 miles of each other must operate on different frequencies. Each frequency costs £50,000 to purchase. How can we go about calculating how many frequencies the operator needs to buy? In the discussion the students will hopefully recognise that we might be able to construct a graph to represent the problem.

What sort of problem is it though? Is it a map colouring problem? Once established that it is, and we already know how to solve these point out this is an example of generalising a problem. But what would we put as nodes and edges? The nodes seem obvious, each base station, but the edges need some thought. We don't need to know about the base stations that are over 150 miles apart. That is unnecessary detail. All we need to represent is the base stations that are within 150 miles. Armed with this knowledge, it should be possible to draw the graph, and apply the same algorithm as before. A student worksheet is included to encourage them to tackle this independently.

We may have algorithms for solving particular classes of problems, but recognising a problem as similar to another is part of the art of computer science. There is no set of rigid rules for solving problems. Recognising that a problem is identical to another is partly experience. Deciding what to include in an abstraction, for example, is a judgement. The ability of students to make sensible judgements, will depend on the intellectual toolkit they develop. That in turn will be based on familiarity with computational concepts that occur again and again, such as abstraction and generalisation. Combining abstraction with algorithms and automating the algorithm through computer programs, (a computational abstraction) is a powerful tool.