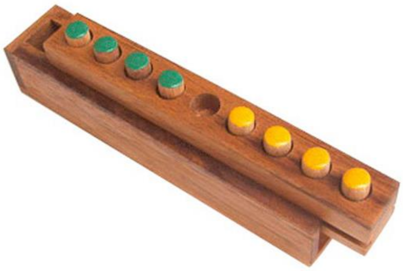
Solving The Peg Swap Puzzle

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

Many children will be familiar with a peg swap (leapfrog) puzzle from their childhood. It is a useful activity to introduce the notion of algorithms and data structures. The practical challenge gives children experience of manipulating values in an array.

Preparation required:
Array and Algorithm sheets + 6 counters (3 of two different colours) for each student.
Wooden puzzles if available.

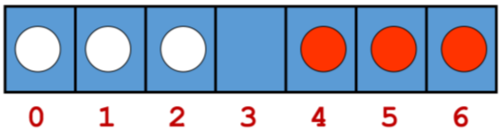
Playing Leapfrog

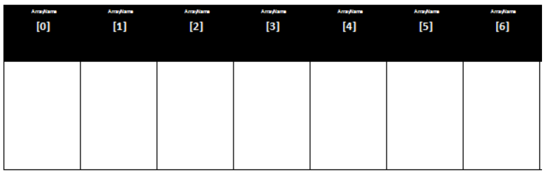


It's likely that the most complex structures a child will manipulate in programs at KS3 will be a list or array. This, in itself is often a difficult conceptual step, children struggling to distinguish between the value in a location and its index position. Before introducing programming challenges, children generally benefit from being exposed to the concept in other ways. Peg puzzles can be a kinaesthetic way into investigating data structures such as arrays. A peg puzzle like this can be made simply with wood and small sections of dowel in two colours. The aim is to swap the pegs, moving each colour alternately. There are two types of move; into an adjacent space or jumping over an opposing colour into a space. These are the only two moves allowed.

The first challenge for students is to find a way to solve the problem. Once some have found a solution, introduce a competitive element as an extension, whilst others are still trying to solve it. What is the minimum number of moves to complete a swap? Children should begin to see a pattern required to complete this efficiently. Pattern recognition like this is a key concept underpinning Computational Thinking. Once students appreciate a pattern, they should be able to extend their reasoning and tackle a larger, similar problem.

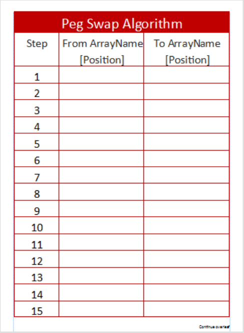
Once all the children have figured out how to solve the smaller puzzle, ask them explain their solution with the puzzle displayed on the board (with index positions initially hidden). Invite them to come up and explain. They may point to pegs initially, but it will become apparent that they need to refer to spaces as well. Discuss possible ways to do this before suggesting (if the children don't) numbering each space. Once we can identify the spaces, do we need to identify the pegs as well? We can articulate an algorithm by recording the move, rather than the peg eg space 4 to space 3.





ArrayName [0]	ArrayName [1]	ArrayName [2]	ArrayName [3]	ArrayName [4]	ArrayName [5]	ArrayName [6]

Hand out an array template, exercise sheet and counters to help them visualise and record their algorithm. If they struggle recording the steps an applet on the cs4fn website (www.cs4fn.org/algorithms/swappuzzle) will record and count the moves as they do them. This can also be useful for a homework extension, perhaps using a larger 9 space board.



Peg Swap Algorithm

Step	From ArrayName [Position]	To ArrayName [Position]
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		

Continue overleaf

The exercise emphasises a way to introduce difficult programming concepts through simple problem solving activities. Often, working things through by hand is an important precursor to any coding. Using an array template to manipulate values helps embed the idea of index positions. There are many other opportunities for a similar approach – developing an algorithm to shuffle cards being one example.