Year 6 Computing — Unit 6.2 — Autumn 1

We are computational thinkers

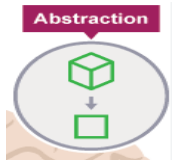
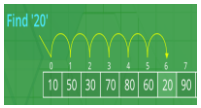
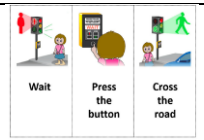
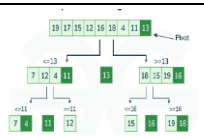
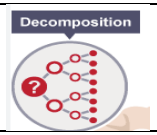
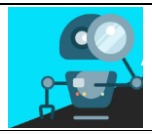
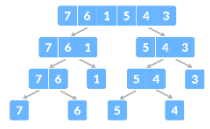
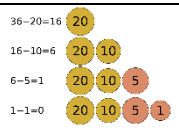
Prior Knowledge

You used presentation software in **Unit 2.4: We are safe researchers**.

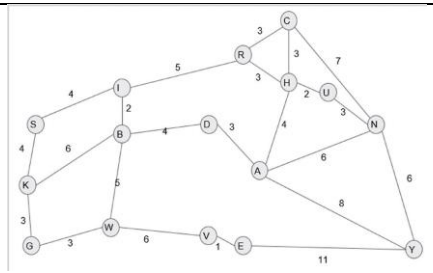
You developed use of presentation software in **Unit 3.3: We are presenters**.

You experienced audio recording in **Unit 3.4: We are who we are** and **Unit 4.3: We are musicians**

Key Vocabulary:

abstraction		computational thinking approach to managing complexity by simplifying things, through identifying what is important and what detail can be hidden or ignored	linear search		search algorithm that looks at each element in turn to see if it meets the criteria
algorithm		a sequence of precise instructions or steps (sometimes a set of rules) to achieve an objective	quicksort:		'divide and conquer' sort algorithm which partitions a list into elements smaller than and larger than a pivot element, and then applies the algorithm to sorting each of these lists
binary search		search algorithm that identifies repeatedly which half of the list of possible elements the target belongs to	search		to identify an element of a list that meets specified criteria
decom-position		breaking a problem down into smaller parts	search algorithm		the way results for a search are selected and ranked, typically through the presence of key words, and by the number and quality of inbound links
divide and conquer		class of algorithms in which the problem is decomposed into smaller, simpler parts, to which the same algorithm can then be applied	selection sort		sort algorithm which looks for the largest element, then the next largest and so on, until the list is in order
graph		data structure showing the connections between elements	sort:		to put a list into order
greedy algorithm		algorithms that work on a 'biggest first' basis, applying divide and conquer methods to reduce the problem rapidly to a simpler problem			

Key learning: How can algorithms be optimised to solve problems quickly?

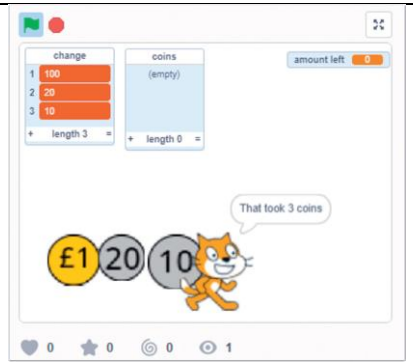


Session 1: Using a network graph

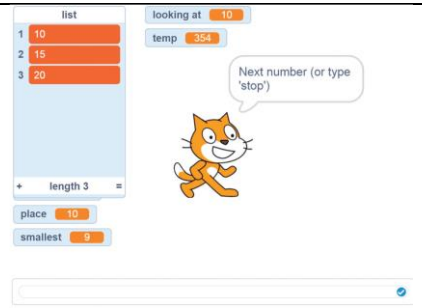
What kind of problems can algorithms be used to solve?

Session 4: Solving a binary search jigsaw in Scratch

How does the Divide and Conquer algorithm work?

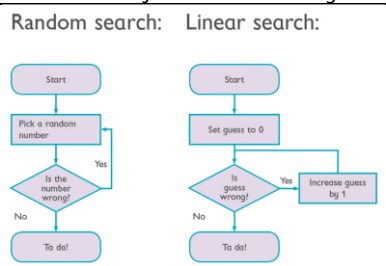


Session 2: A Scratch program to find the smallest number of coins to make a given sum



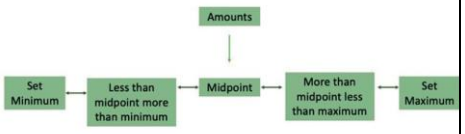
Session 5: Fixing the buggy Scratch selection sort program

How can algorithms be optimised to solve problems more quickly?



Session 3: Exploring random and linear search algorithms

How do I break down problems into smaller parts?



Session 6: Using a quicksort algorithm

Key Applications (apps)

Next Steps

You will be introduced to some of the algorithms for machine learning and other aspects of artificial intelligence in **Unit 6.6: We are AI developers**.