Explain the significance of sorting, in relation to searching.

Define the sorting problem: arranging a list of items in a specific order.

Describe how bubble sort is used for ordering a list of items.

Describe how insertion sort is used for ordering a list of items.

Describe how merge sort is used for ordering a list of items.

Describe how the sorting problem is defined.

Traverse a list of items, swapping the items that are out of order.

Perform a bubble sort to order a list containing sample data.

Perform an insertion sort to order a list containing sample data.

Perform a merge sort to order a list containing sample data.

Interpret and analyse code for bubble sort and insertion sort.

Interpret code for bubble sort and insertion sort with sample data.

Trace code for bubble sort and insertion sort with sample data.

Identify factors that could influence the efficiency of a bubble sort implementation.

Interpret and fix errors in a flowchart.

Describe how bubble sort is used for ordering a list of items.

Describe how insertion sort is used for ordering a list of items.

Describe how merge sort is used for ordering a list of items.

Interpret algorithms and suggest improvements.

Develop a linear search function in Python.

Perform searching and sorting algorithms on samples of data.

Resources are updated regularly - the latest version is available at: the-cc.io/curriculum.

This resource is licenced by the Raspberry Pi Foundation under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International licence. To view a copy of the licence, visit see creativecommons.org/licenses/by-nc-sa/4.0/.