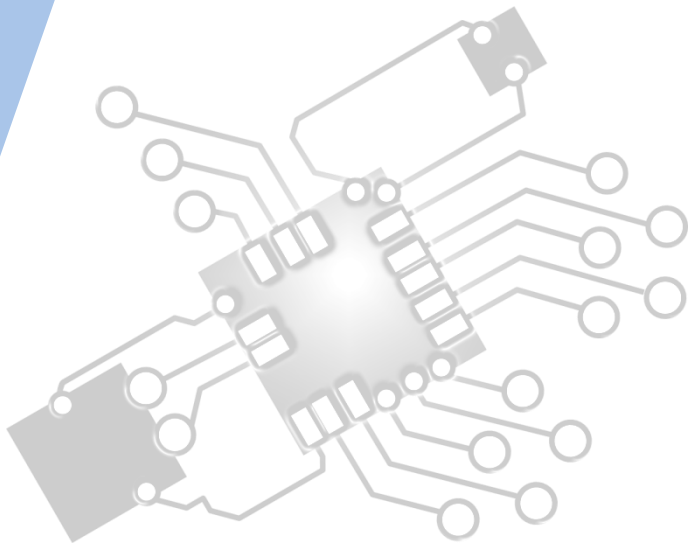
Computational thinking, problem-solving and programming:

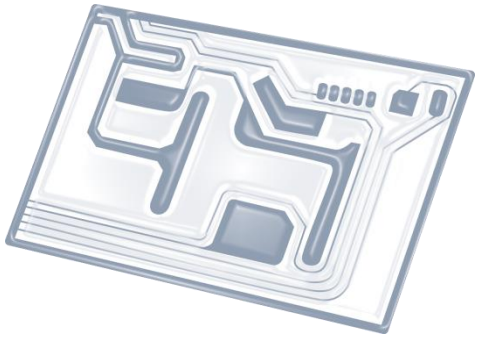
Connecting computational thinking and program design

IB Computer Science



Content developed by
Dartford Grammar School
Computer Science Department

HL Topics 1-7, D1-4



1: System design

2: Computer Organisation

3: Networks

4: Computational thinking

5: Abstract data structures

6: Resource management

7: Control

D: OOP

HL & SL 4.2 Overview

4.2.1 Describe the characteristics of standard algorithms on linear arrays

4.2.2 Outline the standard operations of collections

4.2.3 Discuss an algorithm to solve a specific problem

4.2.4 Analyse an algorithm presented as a flow chart

4.2.5 Analyse an algorithm presented as pseudocode

4.2.6 Construct pseudocode to represent an algorithm

4.2.7 Suggest suitable algorithms to solve a specific problem

4.2.8 Deduce the efficiency of an algorithm in the context of its use

4.2.9 Determine the number of times a step in an algorithm will be performed for given input data

1: System design

2: Computer Organisation

3: Networks

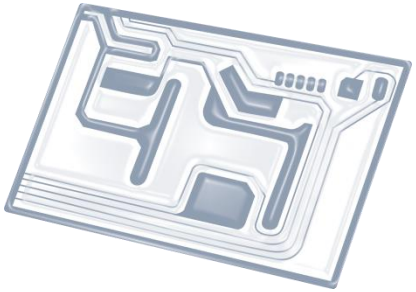
4: Computational thinking

5: Abstract data structures

6: Resource management

7: Control

D: OOP



Topic 4.2.3

Discuss an **algorithm** to solve a specific problem

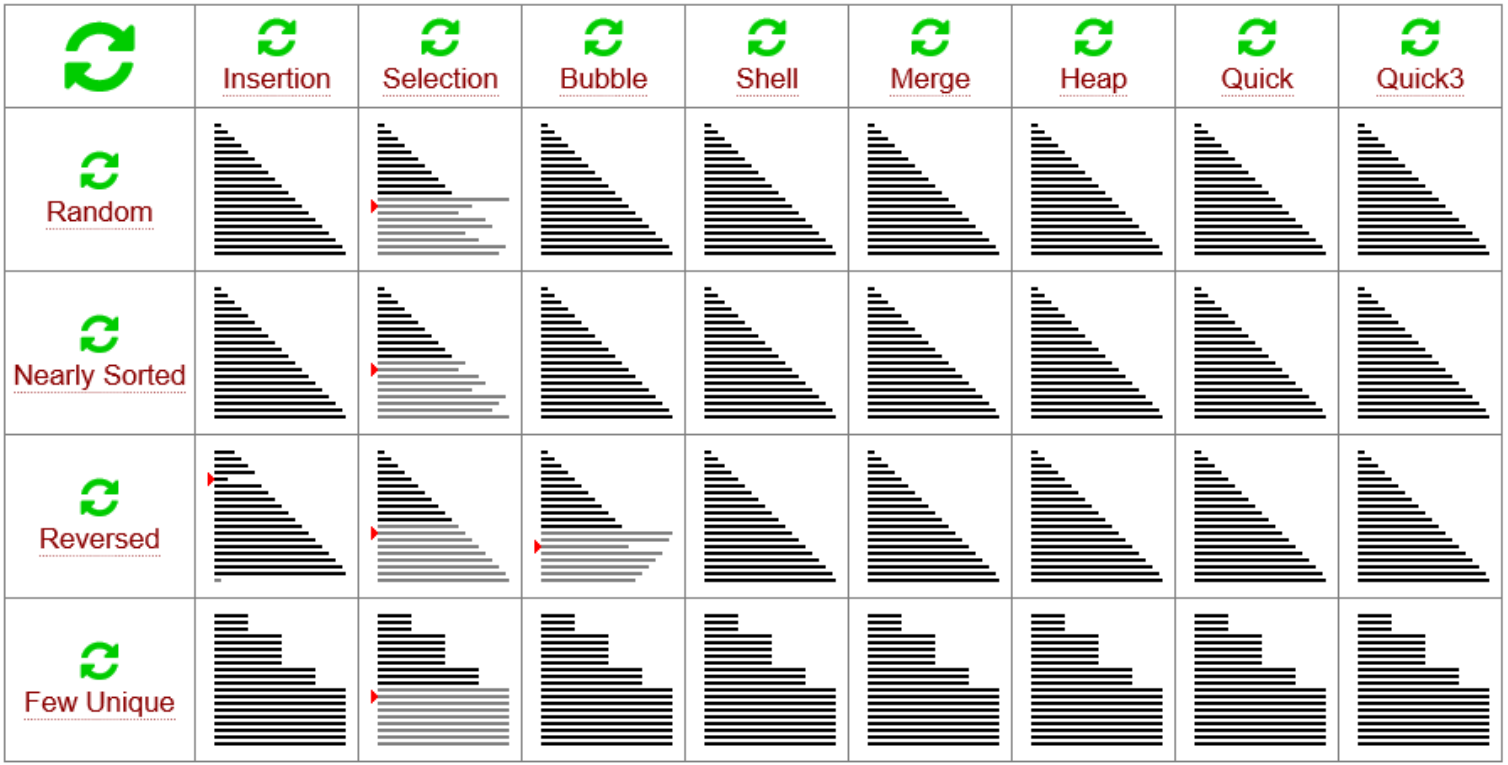
From the teacher's notes:

- *Students should be expected to discuss the differences between algorithms, including both standard and novel algorithms.*
- *For example, discussing the **advantages** and **disadvantages** of using a binary search as opposed to a sequential search.*

Basic comparisons

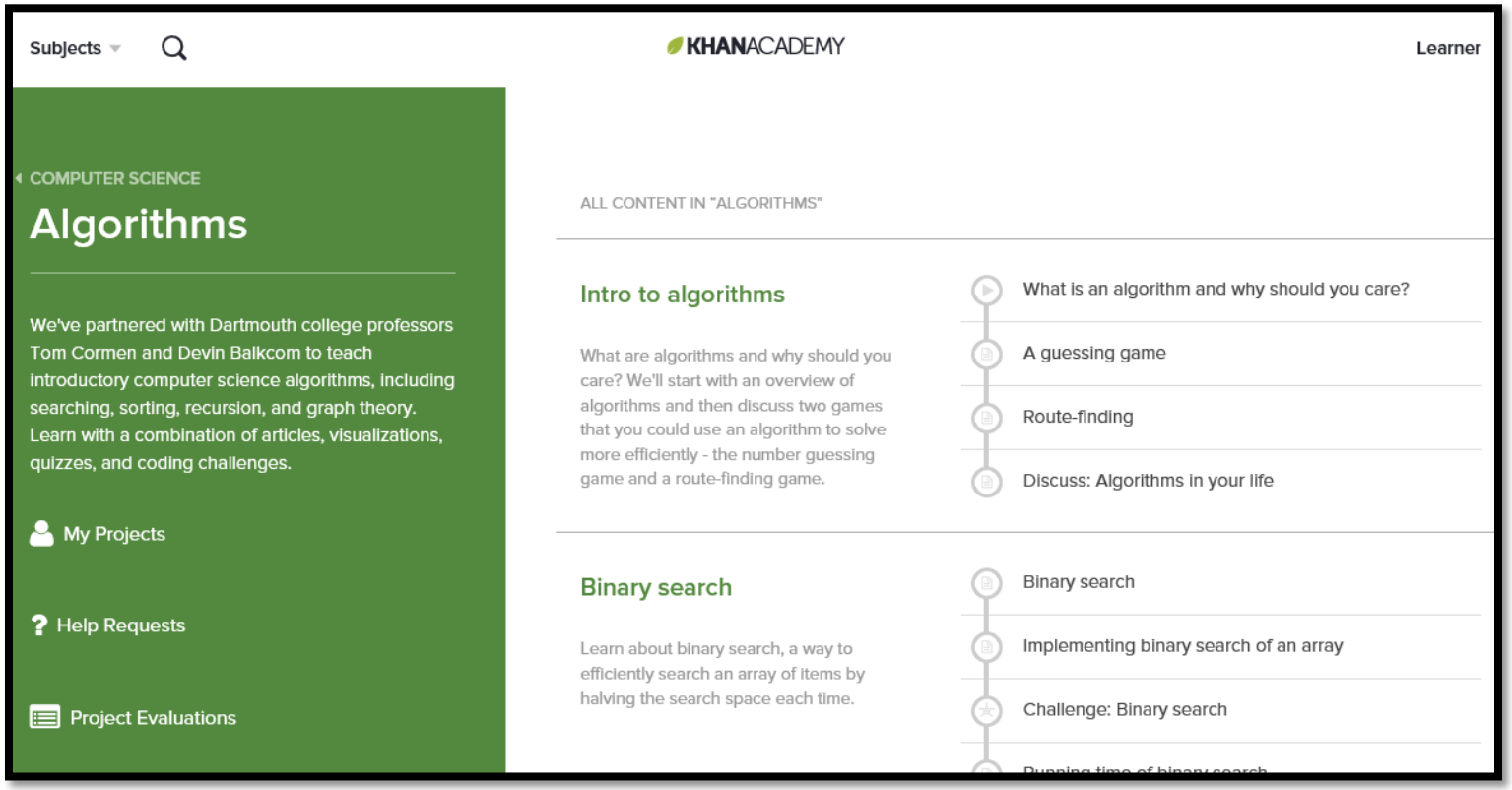
- A **binary search** is **faster** - O(log N),
  but can only be performed in a **SORTED** list
- A **sequential search** is **slower** - O(N)
  but can be performed whether the list is sorted or not
- A **bubble sort** can "quit early" if no swaps are made in a pass. But it makes lots of swaps.
- A **selection sort** must always perform N passes - it cannot "quit early". But it makes fewer swaps - maximum of N swaps
- Both **bubble** and **selection sort** are O(n^2) = equally complex

Useful resources for algorithms



http://www.sorting-algorithms.com/

Useful resources for algorithms



https://www.khanacademy.org/computing/computer-science/algorithms