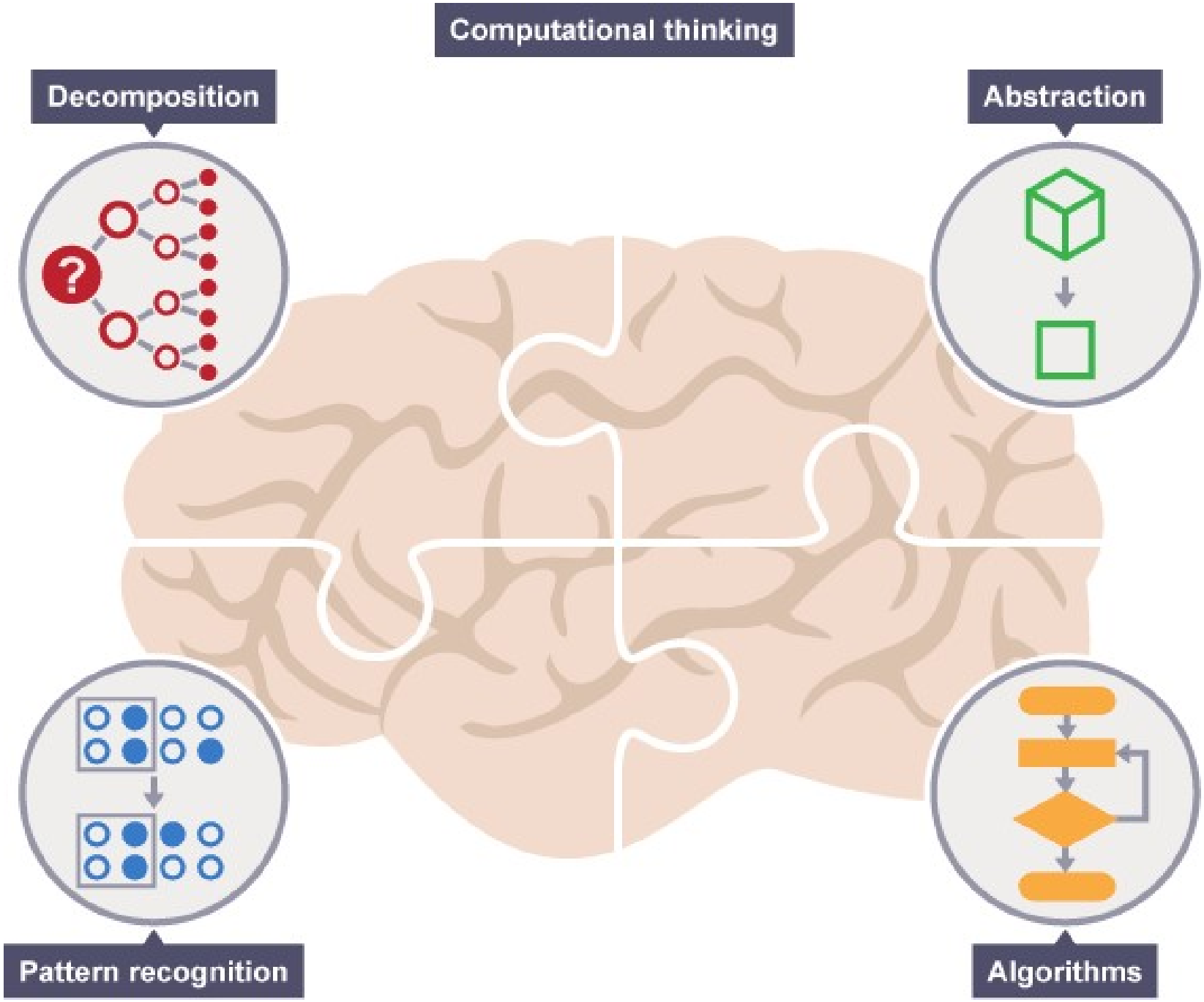
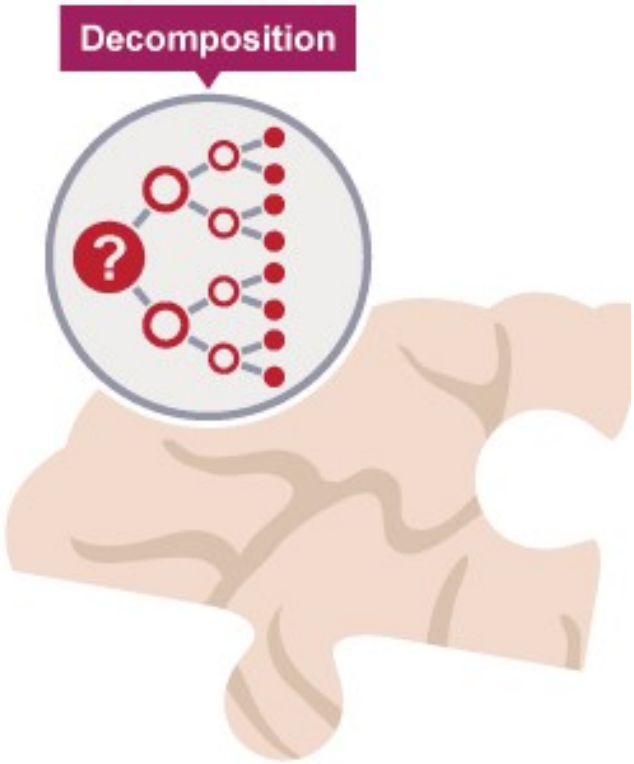
Computational Thinking

An introduction

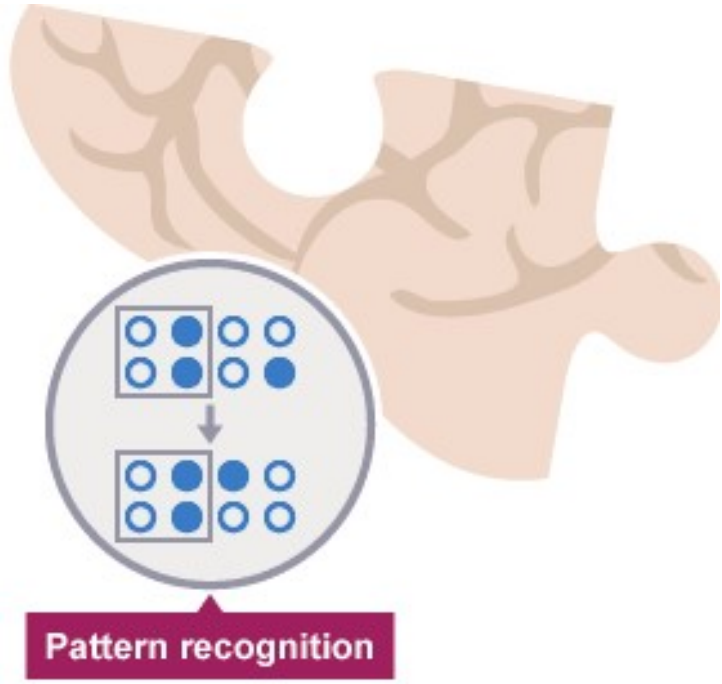

Computational thinking
Decomposition
Abstraction
Pattern recognition
Algorithms

bbc.co.uk



Decomposition

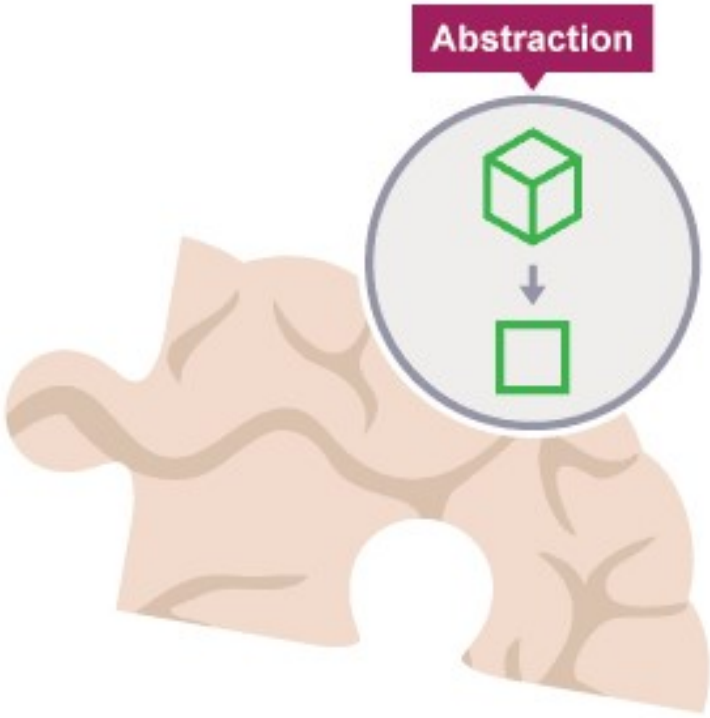
- Breaking a problem down into smaller parts
- Each part could represent a function (a task)
- Easier to work on a large project
  - The different parts are often given to members of a team and later added together



Pattern Recognition

- Noticing a pattern that is repeated in the situation you are trying to model
- More efficient to code if you can repeat sections

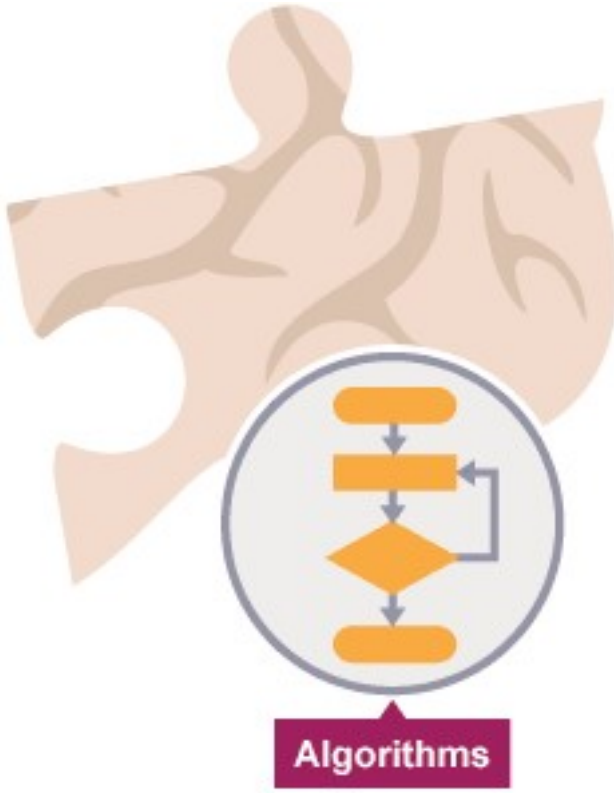
Abstraction



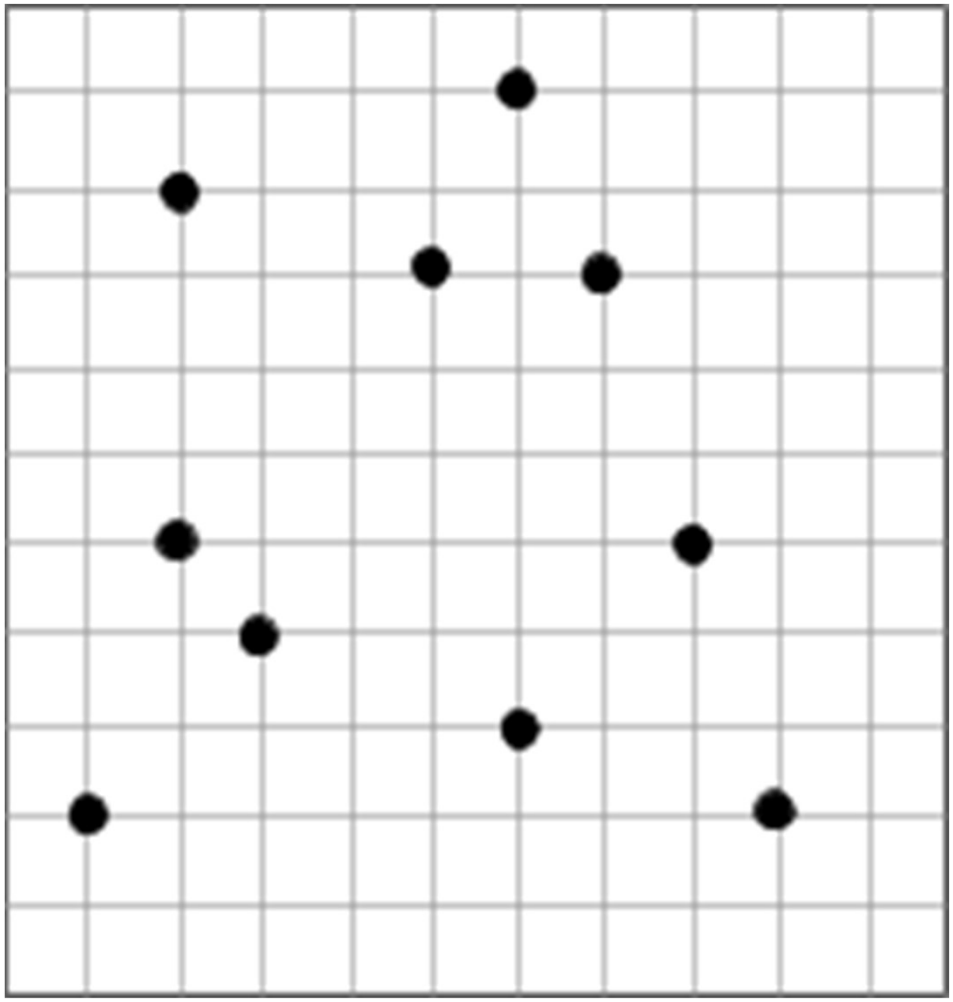
- Seeing the difference between the general and the specific
- General code can then be reused in a specific way

Algorithms

- Working out the individual steps before coding
- If you can solve a problem outside of the coding environment, it then becomes much easier to code

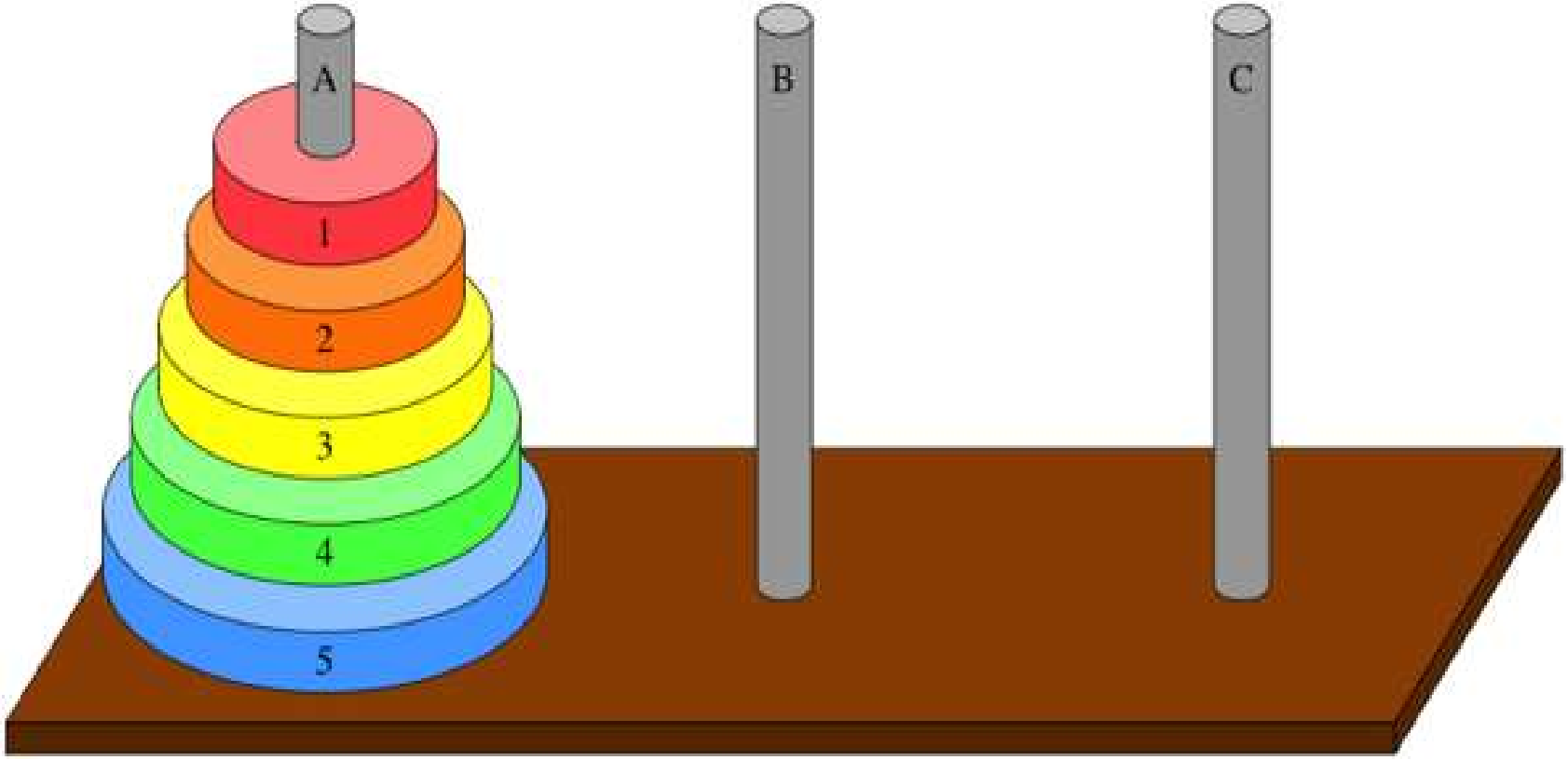


Shortest Path

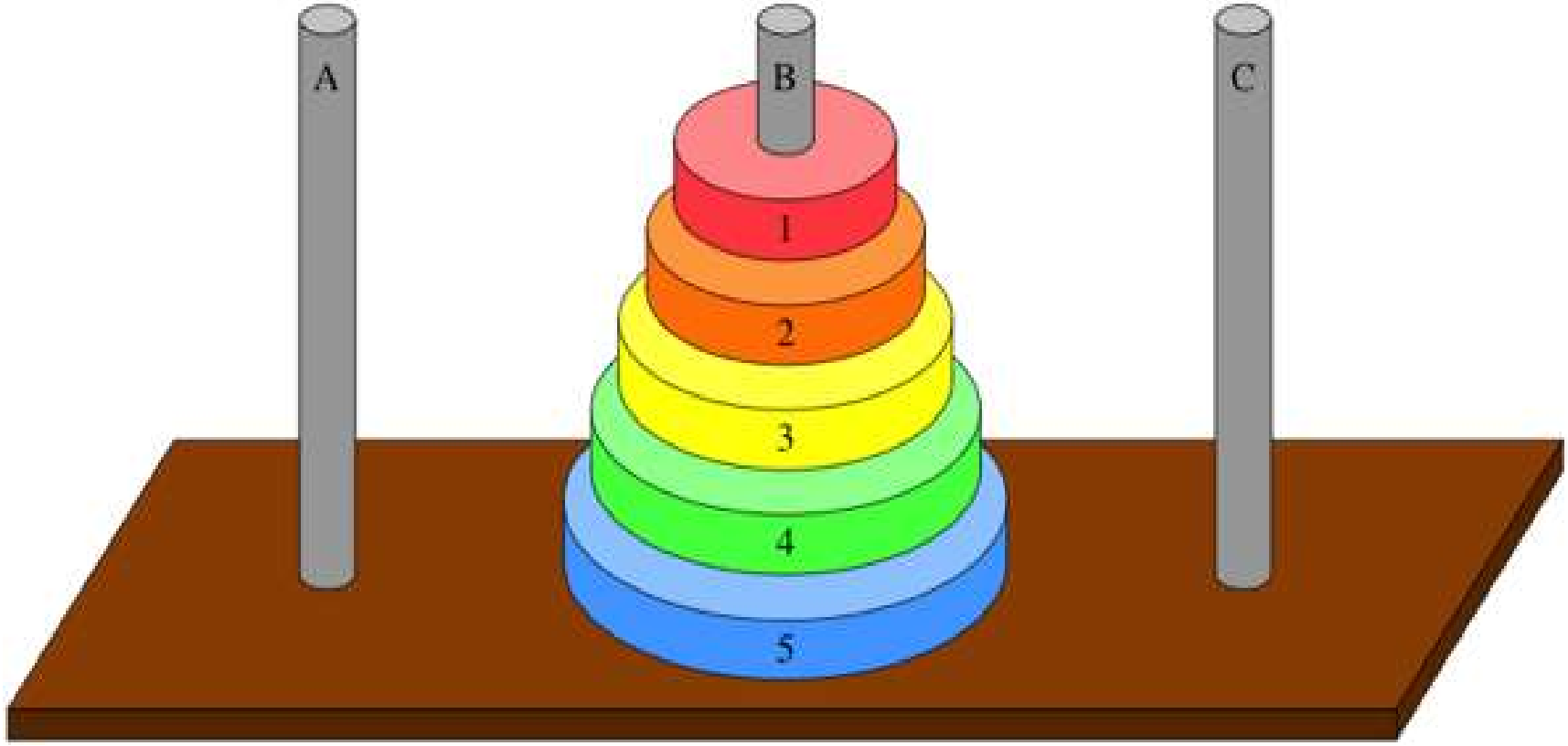


Sides of the cube are 5 meters long

Towers of Hanoi



Intended result

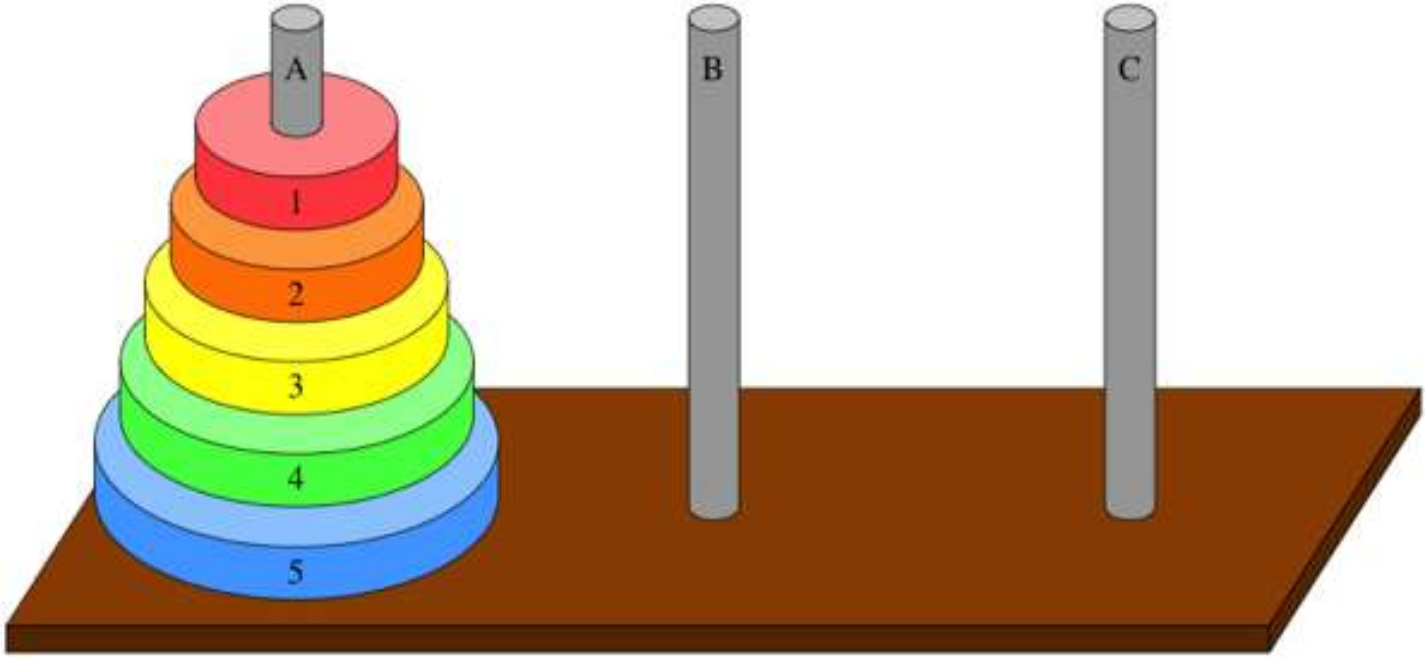


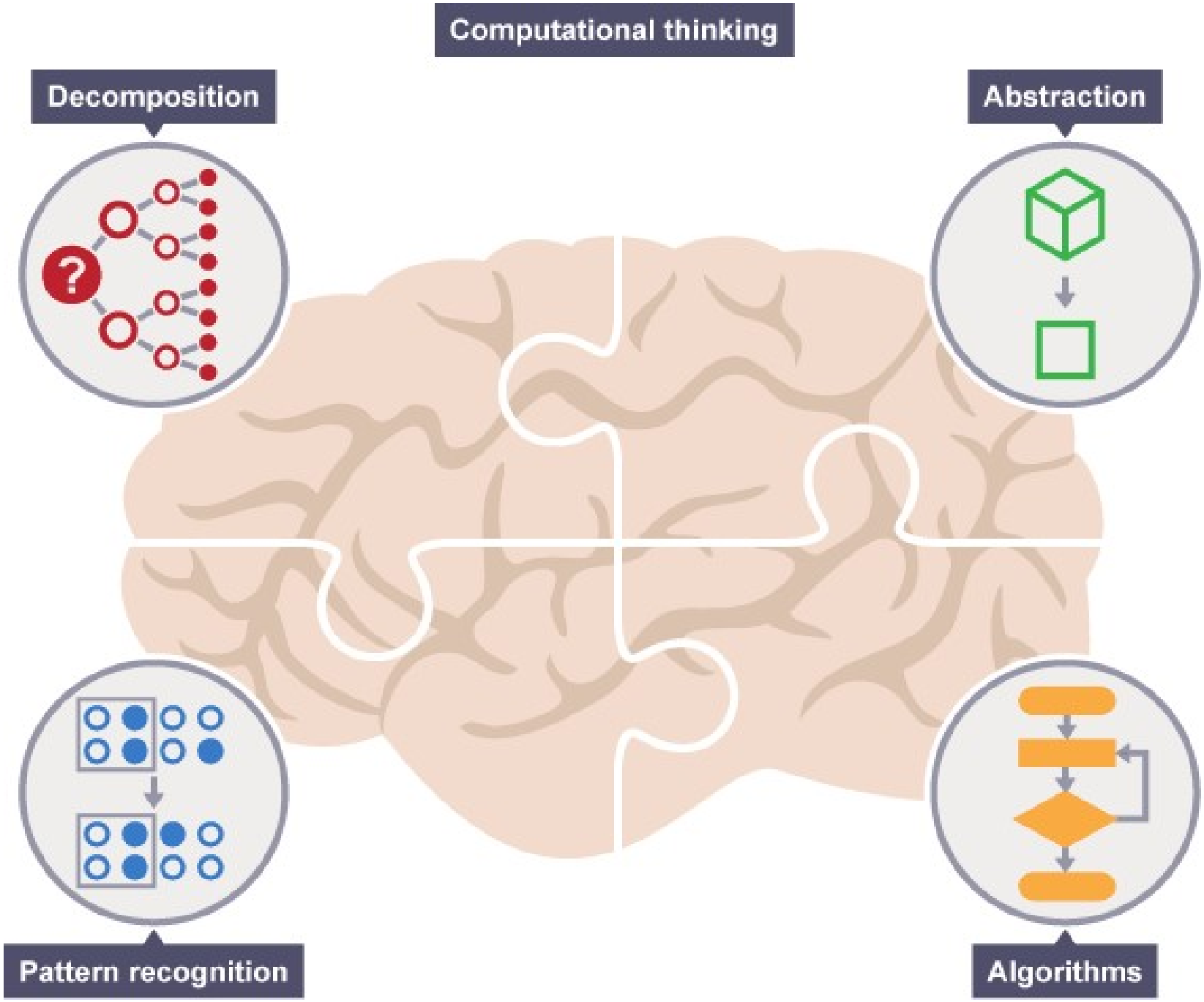
However…

You have to obey two rules:

- You may move only one disk at a time.
- No disk may ever rest atop a smaller disk. For example, if disk 3 is on a peg, then all disks below disk 3 must have numbers greater than 3.

- Following the rules, move all the disks from peg A to B




Computational thinking
Decomposition
Abstraction
Pattern recognition
Algorithms

bbc.co.uk