

# Data structures and algorithms – an introduction

Lecturer: Arran Stewart

# Outline

We discuss:

- ▶ What are data structures? and
- ▶ What are algorithms?

# Algorithm

**algorithm** a clearly specified set of instructions a computer will follow to solve a problem.

# Data

- ▶ Many algorithms require data to be arranged in a specific way if we want the algorithm to run efficiently.
- ▶ This “way of arranging data”, and the *operations* allowed on it (i.e., things we can do to it) are called a *data structure*.

## Example of abstraction

- ▶ We will discuss some real-world examples of abstraction.

# Benefits of studying data structures

- ▶ Many data structures (especially various sorts of *arrays*, *lists*, *trees* and *maps*) are used again and again in software.
- ▶ If you have a good knowledge of them, they form a toolkit of problem-solving methods you can apply to a very wide range of domains.

# Libraries

- ▶ Often, modern languages (like Java and Python) come with many libraries that already include efficient implementations of particular algorithms and data structures.
- ▶ So we don't need to write out own from scratch.
- ▶ But it's *still* important to know the basics of data structures and algorithms, so we know what techniques to use when.

# Summary

We know should know

- ▶ what an algorithm is
- ▶ what a data structure is
- ▶ why we study them