# Software 2 Theory Lecture 1 Algorithms

An algorithm describes a method for solving a problem with some properties: like inputs – zero or more input, output – one or more outputs, termination – finishes in a finite number of steps, definiteness – the description is precise and unambiguous.

## Correctness

An algorithm must be correct in a mathematical sense – for every input, the output should be predictable using the applied algorithm steps.

It’s often easier to show that an algorithm is incorrect.

Various proof techniques -> counter-example (looking for an unusual or extreme case that might not be handled – like testing essentially), induction (prove a base case, and then all other cases following that base case – useful for iterative or recursive algorithms), contradiction (assuming the algorithm is correct, then showing that this leads to a contradictory statement), invariant.