Algorithms: Turing

Key Learning Content

This film describes the motivations of the British mathematician, Alan Turing, famous for his role as a code-breaker in the Second World War, and for building the first computer. Turing wondered whether or not a machine could ever perform as a human being, and built increasingly sophisticated computers to try to answer his question.

The role of algorithms in computer processing is explained and an example is shown on screen. No prior mathematical knowledge is required prior to watching the film, although familiarity with the concept of an algorithm would be useful.

Core Outcomes

Learning Points

- Be able to use and apply number in everyday personal, domestic or community life.
- Be able to understand what an algorithm is, and interpret instructions given in algorithmic form.
- Be able to write simple algorithms to solve problems.

Suggested Activities

- Solve a simple equation by estimation and improvement and document the process using a flow chart.
- Use a calculator to automate the equation-solving algorithm.

Turing's greatest theoretical invention, the Universal Turing Machine, was based on a step-by-step problem-solving process called an algorithm.

Extension Outcomes

Learning Points

- Be able to understand how surds can be approximated using iterative procedures.
- Be able to write simple computer programs.
- Be able to understand the concept of a quadratic expression and be able to factorise such expressions.
- Be able to multiply brackets together containing linear terms in x.
- Be able to manipulate algebraic fractions.

Suggested Activities

- Find the value of a surd by estimation and improvement and document the process using a flow chart.
- Use a calculator to automate the surd-estimation algorithm.
- Write a computer program to estimate surds.

Related Films

To use before the lesson plan:

The Birthday Paradox

This film demonstrates how using equations and algebra can prove that two students in a typical classroom are likely to share the same birthday.

The Arabic Science of Balancing

This film looks at the beginnings of algebra and the man who gave his name to the word ‘algorithm’.

To use after the lesson plan:

Enigma: Cracking the Code

This film describes Turing’s secret work breaking German military codes in the Second World War.

The Greeks and Proof

This film shows how the logic of computer programming owes much to the principles of reasoning laid down by the Greeks over two thousand years ago.

Binary: The Computer Language

This film gives a description of the two-digit mathematics used in every computer ever made.

Guide Lesson Plan

Introduction

Ask the students to use their calculators to work out the square root of two, to as many decimal places as the calculator will give. Then ask: How was this number worked out? Ask them to calculate the square root of a random large number, say 2,473,399; how was this root worked out? How is it programmed into the calculator?

Show Film

Algorithms: Turing

Main Activity

Foundation

Explain that the students are going to work out the square root of two by making successive estimates in a systematic way. To start, they need a number that squares to less than 2, and a number that squares to more than 2, thus: take numbers 1 and 2. Then the square root of two lies between these two numbers. Then take the number midway between 1 and 2 and square it. So 1.5 squared equals 2.25, which is greater than 2. So the square root of two lies between 1 and 1.5. Repeat this process, finding successively better estimates of root two. Get student to repeat the process several times with different starting numbers, and compare estimates with the actual value of root two. Document the process using a flow diagram and explain that the flow diagram describes a mathematical algorithm.

Main Activity cont ...

Advanced

Use a more sophisticated method of estimating root two than described above. Ask students to solve the equation $x^2 = 2$. Rewrite this as follows:

$$x^2 - 1 = 1$$
$$(x-1)(x+1) = 1$$
$$(x-1) = 1/(x+1)$$
$$x = 1/(x+1) + 1$$
$$x = (x+2)/(x+1)$$

so we can then write:

$$x_{n+1} = (x_n+2)/(x_n+1)$$

where xn is our nth estimate of the square root of two.
We can use this formula to work out better and better estimates of root two.

Program this into a calculator as follows: clear the calculator memory, then type 1, then '=', so that 1 is stored in the calculator as 'Ans'. Then enter (Ans+2)/(Ans+1). Press return repeatedly to get better and better estimates of root two. Document this process using a flow chart and explain that this is a mathematical algorithm.

Extension Activity

Using the same approaches as above, adapt the algorithms to work out the square root of any number which is not a square number. So, for example, to find the root of the number k, we use the formula

$$x_{n+1} = (x_n+k)/(x_n+1)$$

Explain that these roots are called surds and that they have infinite, non-recurring decimal expansions.

Optional Extra

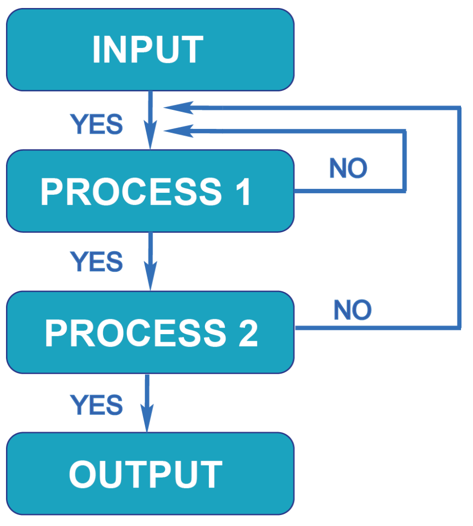
Experiment with the methods described above to solve more complex equations, for example

$$x^3 - 2x^2 + 3x = 4$$

rearranging it as:

$$x^3 = 4 + 2x^2 - 3x \ldots$$

Does the method always work?



An algorithm starts with an input and dictates the specific processes that must be performed to reach a final output.