how hard can it be?

• algorithms:
  • what can they do?
  • what can’t they do?
in three easy parts

• giving instructions

• sequence, parameters

• conditionals, iteration, recursion

• needles in haystacks

• hard to find; easy to recognise

• some things are impossible
Alice laughed: "There's no use trying," she said; "one can't believe impossible things."

"I daresay you haven't had much practice," said the Queen. "When I was younger, I always did it for half an hour a day. Why, sometimes I've believed as many as six impossible things before breakfast."

Alice in Wonderland.
Algorithmic problems

- is 3 prime?
- is 3719 prime?
- is 1024 prime?
- is n prime?
- does $3719^2 = 13,830,961$?
- does $x \times y = z$?
computable questions

- program A - a sequence of instructions
- input D - any text string of data
- A ( D ) - result of program A with input D
- result: true T or false F
- or the program never stops

Alan Turing (1912 – 1954)
Alonzo Church (1903 – 1995)
computable questions

we say “N is prime?” is **computable** because there is a program A such that

- A(N) = T if N is a prime number
- A(N) = F if N is not a prime number

a True/False question is **computable** if there is a program that computes the answer and halts for every input
computable questions

a program A is just a string of text, so can be used as data

is this question computable:

• “Does program A halt with input D?”

if it is computable, there is a program P such that

• P ( A, D ) = T if A halts with input D
• P ( A, D ) = F if A does not halt with input D
(not) halting

• \( P(\mathcal{A}, \mathcal{D}) = \text{Good} \) if \( \mathcal{A} \) halts given input \( \mathcal{D} \)
• \( P(\mathcal{A}, \mathcal{D}) = \text{Bad} \) if \( \mathcal{A} \) does not halt on input \( \mathcal{D} \)

let \( \mathcal{Q} \) be a new program.

\( \mathcal{Q} \) takes an input \( \mathcal{A} \) - then runs \( P(\mathcal{A}, \mathcal{A}) \)
• if \( P(\mathcal{A}, \mathcal{A}) = \text{Bad} \) then \( \mathcal{Q}(\mathcal{A}) \) outputs \( \text{Bad} \)
• if \( P(\mathcal{A}, \mathcal{A}) = \text{Good} \) then \( \mathcal{Q}(\mathcal{A}) \) loops

Saturday, 3 December 2011
(not) halting

- $P(A, A) = \text{Good}$ if $A$ halts given input $A$
- $P(A, A) = \text{Bad}$ if $A$ does not halt given input $A$

let $Q$ be a new program.

$Q$ takes an input $A$ - then runs $P(A, A)$

- if $P(A, A) = \text{Bad}$ then $Q(A)$ halts
- if $P(A, A) = \text{Good}$ then $Q(A)$ loops
(not) halting

- \( P (A, A) = \text{Good} \) if \( A \) halts given input \( A \)
- \( P (A, A) = \text{Bad} \) if \( A \) does not halt given input \( A \)

What's the looping behaviour of \( Q \) run on \( Q \)?

\( Q \) run on \( Q \) - first runs \( P (Q, Q) \)

- If \( \text{Bad} \) then \( Q(Q) \) halts - should be \( \text{Good} \)
- If \( \text{Good} \) \( Q(Q) \) loops - oh dear! what to do?
Author Katharine Gates recently attempted to make a chart of all sexual fetishes.

Little did she know that Russell and Whitehead had already failed at this same task.

Hey, Gödel - we're compiling a comprehensive list of fetishes. What turns you on?

Anything not on your list.

Uh... hm.

Bertrand Russell (1872 – 1970)
• Every provable statement is true.

• Is every true statement provable in arithmetic?

• Kurt Gödel constructed a statement which is true if, and only if, it is not provable in arithmetic. Intuitively it ‘says’

• This statement is not provable in arithmetic - but it is true.