Chapter 1. Section 4

Algorithms

Examples: given in **pseudocode**.

1. An algorithm to determine if x belongs to a given list L of integers:

```
found:= FALSE;
for each element y of L do
    if x = y then found:=TRUE end if
end do;
return found
```

2. An algorithm to list all prime numbers:

```
x := 1;
while TRUE do
    x := x + 1; prime:=TRUE;
    for y := 2 to x − 1 do
        if y | x then prime:=FALSE end if
    end do;
    if prime then print(x) end if
end do
```

3. An algorithm to determine if n is prime:

```
choose a from {2, . . . , n − 1} coprime to n;
if a^(n−1) mod n = 1 then
  return(TRUE)
else
  return(FALSE)
```

Each of the above algorithms raises questions about **efficiency** or **correctness**! Such questions are a major preoccupation of computer scientists; we shall restrict ourselves to mentioning some 'definitional' issues:

Formal vs informal: do the operations in Algorithms 1–3 form 'well-defined sequences'? There are various ways to address this: **Turing machines** or **λ-calculus** are widely used and accepted formalisations. **Church-Turing Hypothesis**: any 'reasonable' model of computation will capture the same class of input-output functions.

Valid inputs and outputs: what might be an **invalid** input for Algorithm 1? Does Algorithm 2 have any inputs? How would you describe its output values?

Terminating vs nonterminating: which of these algorithms is guaranteed to terminate for any input? What if list L in Algorithm 1 is non-finite?

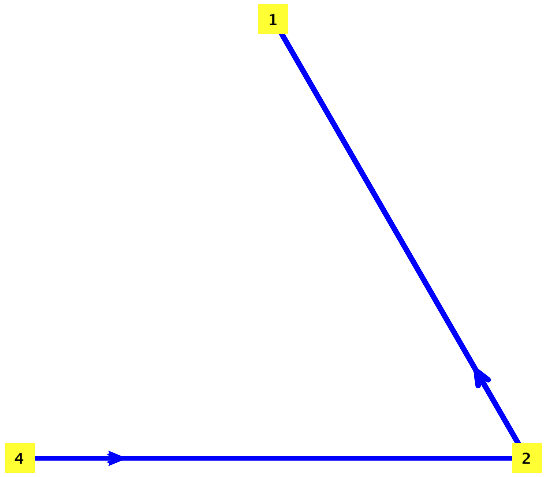
Deterministic vs nondeterministic: will Algorithm 1 always give the same output for any given input x? What about Algorithm 3? What issue does the word 'choose' involve in Algorithm 3?

A Graph Theory Example

The following algorithm contains each of the elements from the table above (with indentation replacing the words **end if** and **end do**). It is conjectured that the algorithm will terminate with output 1 for any positive integer input but this is a notorious unsolved problem, sometimes called the Collatz Conjecture. A whole book has recently been published on the subject: *The Ultimate Challenge: The 3x + 1 Problem*, J.C. Lagarias (ed.), AMS, 2011.

```
Collatz(x)
# Input: a positive integer x
        while x ≠ 1 do
            if x is even then x := x/2
            else x := 3x + 1;
        return(x)
```

We can represent the action of the Collatz algorithm as a digraph. It has an infinite set of vertices: the positive integers; and there is an edge from x to y if $y = x/2$ or x is an odd number greater than 1 and $y = 3x + 1$. The subgraph induced by vertices 1, 2 and 4 is shown on the right: it is an **in-tree** with root 1.



We can add more and more vertices: the smallest subgraph which contains all the vertices $1, \ldots, 26$ is shown on the next page. This subgraph has 41 vertices. Again it is an in-tree rooted at 1, although this is much harder to see now! So it has $41 - 1 = 40$ directed edges and is connected, with every vertex having a directed path to 1.

The behaviour of the Collatz digraph is very hard to predict. For example, the vertex 27 is missing from the above subgraph. The smallest subgraph containing vertex 27 has 135 vertices, because it takes Collatz(27) many steps to reduce 27 to 1. This corresponds to a long path from vertex 27 to vertex 1 in the digraph. In fact this path has 112 vertices and passes by vertex 9232 on its route!

We can formulate many unanswered questions about the so-called $3x + 1$ problem in terms of our digraph:

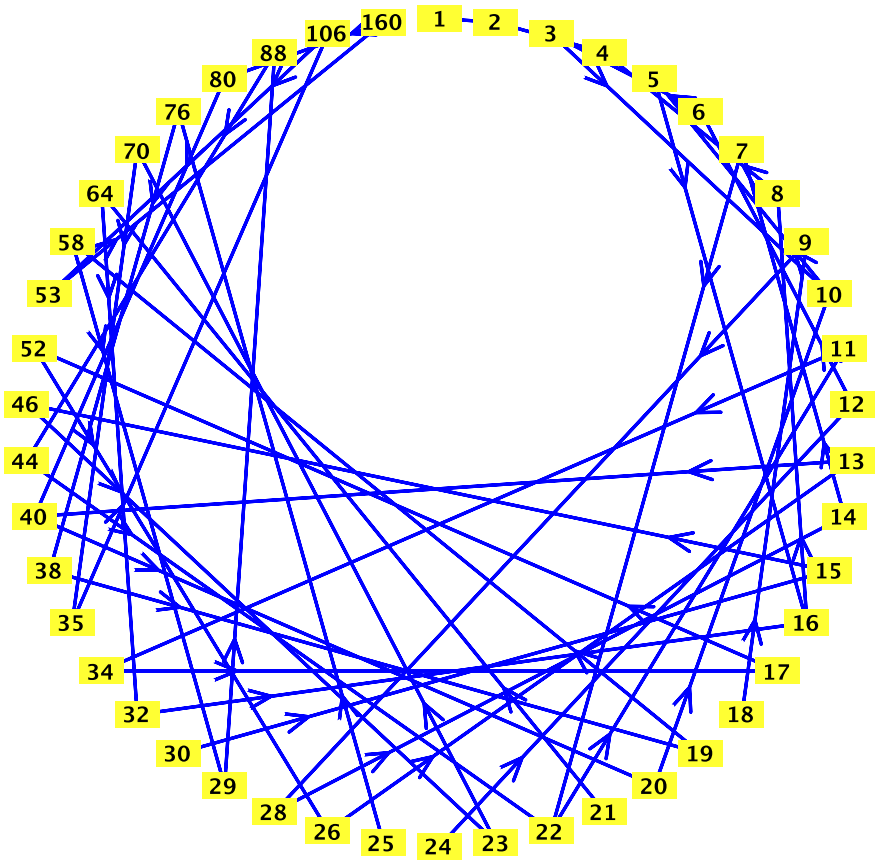
1. Does it contain any directed cycles? If we allow an edge out of vertex 1 then this would go to 4. Would this be the only cycle? If we allow negative inputs then there is a cycle $-5 \to -14 \to -7 \to -20 \to -10 \to -5$. So how does the digraph behave generally for negative integers?

2. Is the graph connected? This is the not the same question: it may be that there are no cycles in the positive integers but that some input creates an infinite path, meeting larger and larger integers. So this would be an infinite component of the graph.

3. If the digraph is not connected then how many components does it contain? Is this number finite or infinite?

4. Can we find a formula for the undirected distance between any two vertices? For instance, we can walk from 12 to 13 in 6 edges, ignoring directions; but the walk from 26 to 27 will be *much* longer! Can we even find bounds for this distance? (A finite upper bound for every pair of positive integers would prove the Collatz Conjecture, of course).

As far as I know, none of these questions has any immediate prospect of being answered!


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
28
29
30
32
34
35
38
40
44
46
52
53
58
64
70
76
80
88
106
160