

ALGORITHMS

An **algorithm** is a finite sequence of well-defined and logical steps used to complete a task, usually to solve a problem. An algorithm has the following properties:

1. No instructions can be dependent on the user. All ambiguity in the process must be removed.
2. The process must end with a solution after a finite number of steps.
3. The process is deterministic; that is, each step unambiguously determines what the next step will be.

Simply stated, an algorithm must receive an input and through a finite process produce a solution. Using the same input value will always produce the same solution (unless a step incorporates randomizing).

An algorithm can have any sort of notation. An algorithm on the GRE could use any of the symbols commonly used in the section on Logic. Typically, ETS uses algorithms expressed in a programming language (or a flow chart). This language is designed to express the steps, or computations, that tell the computer how to behave. The syntax of a programming language used on the GRE should be simple enough to be understood by any test taker who has taken an undergraduate course in computer science.

Example 7.6 The following algorithm is used to prove the famous “ $3x + 1$ ” problem.

```
input (a)
while a > 1
  begin
    if (a mod 2 = 0)
      a := a / 2
    else a := 3 • a + 1
    output (a)
  end
```

When the input value of the algorithm is 3, what is the sequence of outputs produced?

Solution: Interpret the steps of the algorithm. The input value is a , and a loop is created while $a > 1$. For each step, if $a = 0 \pmod{2}$, then divide a by 2. Otherwise multiply a by 3 and add 1. Finally, the algorithm will output the new value of a and reevaluate whether $a > 1$ before continuing.

The outputs of the algorithm will be the following:

10, 5, 16, 8, 4, 2, 1

Because a is not greater than 1, the process ends, and the sequence converges to a solution.

and three