What's an algorithm?

A precise definition of **algorithm** is difficult, which is interesting since the concept has become central to much of mathematics and computer science during the last quarter century. It is as vital and important to such study as the **sonnet** is to the history and practice of poetry. Here are some quotes from Knuth's *The Art of Computer Programming*.

From page 1:

> The word "algorithm" itself is quite interesting; at first glance it may look at though someone intended to write "logarithm" but jumbled up the first four letters. ... the true origin of the word ... comes from the name of a famous Persian textbook author, Abu Ja'far Mohammed ibn Mûsâ al-Khowârismî (*c.* 825) – literally, "father of Ja'far, Mohammed, son of Moses, native of Khowârizm." Khowârizm is today the small Soviet city of Khiva. Al-Khowârizmî wrote the celebrated book *Kitab al jabr w'al-muqabala* ("Rules of restoration and reduction"); another word, "algebra", stems from the title of his book, although the book wasn't really very algebraic.

From pages 4, 5, and 6:

> The modern meaning for algorithm is quite similar to that of *recipe, process, method, technique, procedure, routine,* except that the word "algorithm" connotes something just a little different. Besides merely being a finite set of rules which gives a sequence of operations for solving a specific type of problem, an algorithm has five important features:
>
> **1) Finiteness.** An algorithm must always terminate after a finite number of steps. ...
>
> **2) Definiteness.** Each step of an algorithm must be precisely defined; the actions to be carried out must be rigorously and unambiguously specified for each case. ...
>
> **3) Input.** An algorithm has zero or more inputs, i.e., quantities which are given to it initially before the algorithm begins. These inputs are taken from specified sets of objects. ...
>
> **4) Output.** An algorithm has one or more outputs, i.e., quantities which have a specified relation to the inputs. ...
>
> **5) Effectiveness.** An algorithm is also generally expected to be *effective.* This means that all of the operations to be performed in the algorithm must be sufficiently basic that they can in principle be done exactly and in a finite length of time ...

Knuth continues on the same page to contrast his definition of algorithm with what could be found in a cookbook:

> Let us try to compare the concept of an algorithm with that of a cookbook recipe: A recipe presumably has the qualities of finiteness (although it is said that a watched pot never boils), input (eggs, flour, etc.) and output (TV dinner, etc.) but notoriously lacks definiteness. There are frequently cases in which the definiteness is missing, e.g., "Add a dash of salt." A "dash" is defined as "less than $\frac{1}{8}$ teaspoon"; salt is perhaps well enough defined; but where should the salt be added (on top, side, etc.)? ...

He concludes his comparison by writing:

> ... a computer programmer can learn much by studying a good recipe book.