In [computer science](https://en.wikipedia.org/wiki/Computer_science), a **deterministic algorithm** is an [algorithm](https://en.wikipedia.org/wiki/Algorithm) which, given a particular input, will always produce the same output, with the underlying machine always passing through the same sequence of states. Deterministic algorithms are by far the most studied and familiar kind of algorithm, as well as one of the most practical, since they can be run on real machines efficiently.

Formally, a deterministic algorithm computes a [mathematical function](https://en.wikipedia.org/wiki/Function_(mathematics)); a function has a unique value for any input in its [domain](https://en.wikipedia.org/wiki/Function_domain), and the algorithm is a process that produces this particular value as output.

Deterministic algorithms can be defined in terms of a [state machine](https://en.wikipedia.org/wiki/State_machine): a *state* describes what a machine is doing at a particular instant in time. State machines pass in a discrete manner from one state to another. Just after we enter the input, the machine is in its*initial state* or *start state*. If the machine is deterministic, this means that from this point onwards, its current state determines what its next state will be; its course through the set of states is predetermined. Note that a machine can be deterministic and still never stop or finish, and therefore fail to deliver a result.

Examples of particular [abstract machines](https://en.wikipedia.org/wiki/Abstract_machine) which are deterministic include the [deterministic Turing machine](https://en.wikipedia.org/wiki/Deterministic_Turing_machine) and [deterministic finite automaton](https://en.wikipedia.org/wiki/Deterministic_finite_automaton).