The **Lamport timestamp** algorithm is a simple [logical clock algorithm](https://en.wikipedia.org/wiki/Logical_clock_algorithm) used to determine the order of events in a [distributed computer system](https://en.wikipedia.org/wiki/Distributed_computer_system). As different nodes or processes will typically not be perfectly synchronized, this algorithm is used to provide a [partial ordering](https://en.wikipedia.org/wiki/Partially_ordered_set) of events with minimal overhead, and conceptually provide a starting point for the more advanced [vector clock](https://en.wikipedia.org/wiki/Vector_clock) method. The algorithm is named after its creator, [Leslie Lamport](https://en.wikipedia.org/wiki/Leslie_Lamport).

*\*happened-before* ordering

start(Name, Logger, Seed, Sleep, Jitter, TimeModule)

Name: The worker is given a unique name and access to the logger.

Logger: To keep track of what is happening and in what order things are done

we send a log entry to the logger every time we send or receive a message.

Seed: We also provide the worker with a unique value to seed its random generator. If

all workers started with the same random generator they would be more in

sync, more predictable and less fun.

Sleep: the sleep value will determine how active the worker is sending messages

Jitter: time delay