Algorithms understanding and design is more than a game, well sorry, it could be in some extent considered the game of life, the intrinsic intelligence of nature to create the mystery of the old Taoist “Ten Thousand Things”. Given an algorithm running it forward is creation of beings that evolve sometimes from “seeds” sometimes from other beings, belonging to the same “family”. As time folds onward it may be most times unfolded backwards either pseudo random or deterministic. My passion by algorithms began reading The Growth and Form, the milestone work of D'Arcy Thompson, for many the first bio mathematician, when I was a teenager. His ideas about gnomes and gnomic transformations were in my humble opinion precursor of fractals.
**Selected Authorities**

We show here our suggested Basic Websites List to initiate your algorithmic navigation either for pleasure or deepening in the art of their understanding and design.

1. **Math Forum**, a well known math forum where you may find basic answers and guides to your questions: [http://mathforum.org/library/drmath/sets/college_algorithms.html](http://mathforum.org/library/drmath/sets/college_algorithms.html). You may find here answers to problems at elementary School, Middle School, High School, and College & Beyond levels. See as an example the meaningful answer about The Euclidean Algorithm: [http://mathforum.org/library/drmath/view/51447.html](http://mathforum.org/library/drmath/view/51447.html).

2. **Dictionary of Algorithms and Data Structures**


The Dictionary of Algorithms and Data Structures is an obliged reference for many of algorithms, "algorithmic techniques", "archetypal problems" and data structures. It is maintained by Paul E. Black, and is hosted by the Software Quality Group of the Software Diagnostics and Conformance Testing Division, Information Technology Laboratory, a part of the [National Institute of Standards and Technology](http://www.nist.gov/dads/).


4. **Data Structures and Algorithms**


I) About Knowledge Management Applications
Here we show a series of images pertaining to Darwin, an AI multi-agent methodology oriented to unveil the “hidden” order of huge Data Reservoirs as the Web. Most algorithms work on High Order Sparse Binary Matrices defined over a GF2 three-dimensional space: 10 billionx10 millionx500,000.

Mapping the Human Knowledge

This figure depicts a complex multi agent task: mapping the knowledge of a huge Data Reservoir (for instance the Web). Resembling a DNA algorithm agents retrieve and discriminate all meaningful “concepts” out from million of documents of a given discipline by detecting clusters of them. Along its semantic skeleton they inspect the semantic neighborhood of any subject. See it at www.intag.org.

How to preserve the "Specificity Rule"

Detail of a sub algorithm performed by agents in the figure above. Given a “flat” unstructured Thesaurus (or Catalog) agents are settled and tuned up to build autonomously and automatically the “hidden” hierarchy, level by level going upwards. Agents match make keywords versus subjects preserving the “Specificity Rule” a sort of entropy preservation.
This algorithm works on one of these GF2 matrices building a weighted spectrum of rather "exceptional" sequences of words/symbols, candidates to become concepts belonging to a given Major Subject mapped.

This is another discrimination algorithm working on pure txt data samples, supposedly "written" with a common Jargon, growing exponentially until all that remains should be the Jargon itself.
This algorithm works on “users’ side” trying to detect most frequent users’ behaviors/traits. Supposedly users intent to reduce their “uncertainty level” about something querying a given Cognitive Offer (for example via a Search Engine). Their queries are instrumented as “session strings” of guessed “keywords” k and navigation instances, all symbols detected through a “Darwin e-membrane” ®. Agents working on the “ectoderm” side of it discriminate potential “users’ keywords” (not the “established” keywords of the queried Knowledge Database) in monads, dyads, triads and n-ads, trying to build versions of a Users’ Thesaurus.

Within a multi-agent platform, a bp-bot, behavior pattern agent, classify users’ sessions within pre-established PAG’s “groups”, characterized by very specific subjects semantic “Darwin fingerprints” ®, maintaining existing and suggesting new ones.
This figure shows the work of a Catalog Items procurement agent, p-bot, within a multi agent platform, performing some basic Catalog Harmonization tasks. Semantically it builds a Thesaurus with a “flat”, unstructured only one level Catalog in as many levels as intrinsically exist. See it working at www.procurebot.com. Agents may work on both sides: classically, on “proprietary” side, and on “users’ side”, trying to understand and aid users daily operations.

This is the scheme of an “anthropic” multi-agent platform managed by KM Darwin Methodology ®. A given task “something to do” is solved via an anthropic Cosmo vision, “half humans”, “half agents”, doing each one what perform best. The purpose of the learning cycle is to transfer to agents as much as possible and as soon as possible, transferring “mechanical” talent to agents. We have experimented with Darwin and our conclusion is that what’s computable, no matter its level of complexity could be successfully transferred to agents. Of course the ultimate control remains always in Human hands!
II) About a Telecom Application

Here we show a series of algorithms working on "flow streams" of digitalized signals, that have to be coded-decoded and errors detected-corrected along their way from origins to destinations. Some of these streams experiment hundreds of transformations via "states' machines" like the one depicted below with five algorithms working synchronically under the control of a "main" program.

This algorithm deals with a rather complex backwards navigation throughout a "states machine", specifically one based on a set of Non linear feedback registers. The core of the algorithm is a Markovian "critical" process.

This algorithm shows a detail of the algorithm depicted above: navigating "upwards" a tree, the process when a node has no "ancestors". It is also shown how the tree is browsed via a special "triplet" sub algorithm.
Navigation by triplets

This algorithm belongs to a higher level of the one shown above. Instead of browsing a 3D "states machine" binary space via "words arrays" performing it via "bits arrays". Each "triplet" identifies state sequences either going forwards or backwards in the most efficient way.

We show here a zoom of above mentioned "navigation via triplets". This is not the place to explain how this algorithm functions. The details are programming oriented, as a "data sample" to aid debugging.
Interleaving

Interleaving stands for a bits permutation within a given data block. Here is depicted the permutation to be performed via table lookups either to interleave or de-interleave.

Trellis - States

A classic Trellis algorithm display applied to a 16 states machine

A 4 states machine Trellis going forwards step

A peculiar States Machine graph, used for example for Convolution coding.

A data sample for a script that detects parasitic cycles within pseudo random virtual machines.