Algorithm design

From Wikipedia, the free encyclopedia

Algorithm design is a specific method to create a mathematical process in solving problems. Applied algorithm design is algorithm engineering.

Algorithm design is identified and incorporated into many solution theories of operation research, such as dynamic programming and divide-and-conquer. Techniques for designing and implementing algorithm designs are algorithm design patterns,[1] such as template method pattern and decorator pattern, and uses of data structures, and name and sort lists. Some current day uses of algorithm design can be found in internet retrieval processes of web crawling, packet routing and caching.

Mainframe programming languages such as ALGOL (for *Algo*rithmic *l*anguage), FORTRAN, COBOL, PL/I, SAIL, and SNOBOL are computing tools to implement an "algorithm design"... but, an "algorithm design" (a/d) is not a language. An a/d can be a hand written process, e.g. set of equations, a series of mechanical processes done by hand, an analog piece of equipment, or a digital process and/or processor.

One of the most important aspects of algorithm design is creating an algorithm that has an efficient runtime, also known as its Big O.

Steps in development of Algorithms

1. Problem definition
2. Development of a model
3. Specification of Algorithm
4. Designing an Algorithm
5. Checking the correctness of Algorithm
6. Analysis of Algorithm
7. Implementation of Algorithm
8. Program testing
9. Documentation Preparation

Common design paradigms [edit]

- Divide and conquer
- Dynamic programming
- Greedy algorithm
- Back Tracking
- Brute Force