DISCRETE MATHEMATICS
Knowledge has Organizing Power

Theme 3
RECURRENCE RELATIONS AND COUNTING
Self-Referral Dynamics

LESSON 7
ALGORITHMS
Using the Organizing Power of Creative Intelligence

Wholeness of the Lesson

Algorithms are the organizing power that allow computers to solve problems systematically, correctly, and efficiently. Creating an algorithm requires knowledge, intelligence, and creativity. *Science and Technology of Consciousness:* Our practice of the Transcendental Meditation technique strengthens the organizing power of our own creative intelligence.

Main Points

1. An algorithm is a finite sequence of precise instructions for performing a computation or solving a problem. The best algorithms are correct, precise, and efficient.
   Science and Technology of Consciousness: The Transcendental Meditation technique is easy, efficient, and systematic, and regular practice leads to success in life.

2. The complexity of an algorithm is the number of operations it takes to reach the answer. It is usually expressed as a function $f(n)$ of the size n of the problem.
   Science and Technology of Consciousness: Our regular practice of the Transcendental Meditation technique locates pure consciousness as the source of all complexity in our daily life.

3. When choosing between algorithms, we choose the algorithm whose size grows the slowest as the size of the problem increases. To make this easy, we compare the growth of each algorithm with a familiar function whose rate of growth is known (for example, a logarithmic, linear, power, exponential, or factorial function) using Big O notation.
   Science and Technology of Consciousness: Growth is natural to life. But growth must be supported by stability. Stability is strengthened by regularly contacting that unchanging field of creative intelligence within us.

UNITY CHART

Connecting the Parts of Knowledge with the Wholeness of Knowledge

Algorithms

1. An algorithm is a systematic procedure for solving a problem.

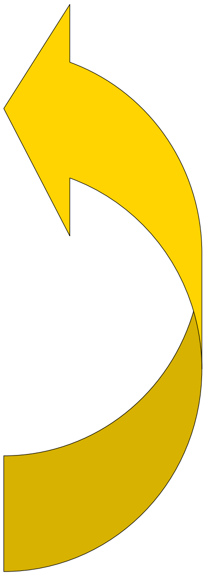
2. Algorithms allow us to harness the enormous power of computers for efficiently solving a wide range of problems.

— — — — — — — — — — — — — — — —

3. **Transcendental Consciousness** is a field of pure knowledge and infinite organizing power.

4. **Impulses within the transcendental field:** structure the relative world using the infinite organizing power of Transcendental Consciousness.

5. **Wholeness moving within itself:** In Unity Consciousness, we harness the infinite organizing power of our own Self to live fulfillment in life.



An algorithm should have the following:

Problem You must have a clear description of the problem to be solved. You must know exactly what is given and what is being requested.

Input You must specify the nature of the input. Your algorithm is expected to work only for input of the specified type.

Output The algorithm gives an output of the appropriate form for every input. The nature of the output is should be clear from the statement of the problem.

Definiteness You must give a precise and unambiguous description of each step of the algorithm. It must be possible to perform each step of the algorithm no matter what the input.

Effectiveness You must be able to perform each step of the algorithm yourself.

Correctness An algorithm must give the correct answer for all possible inputs. You may have to prove that the algorithm does give the correct answer.

Finiteness No matter what the input, the algorithm must give an answer after a finite number of steps. Each step must take a finite time for execution.

Generality The algorithm must be applicable to all cases of the problem to be solved.

Types of Algorithms

1. Greedy algorithms: for optimization problems, at each stage make the best possible choice. The nature of life is to grow towards more and more.

2. Brute-force algorithms: compute all possible outcomes. For effective action, function from the field of all possibilities.

3. Recursive algorithms: perform the same step over and over, with simpler inputs each time. Creation from within is a process of self-referral.

4. Backtracking algorithms: use depth-first search on a graph or tree. By transcending, we reach the home of all knowledge.

5. Divide and conquer algorithms: divide the problem into smaller problems of the same type, solve these problems, and combine the results into one solution of the original problem. The whole is contained in each part.

6. Dynamic programming algorithms: for optimization problems, remember past values to help find better values. Progress is through steps of rest and activity.

7. Branch and bound algorithms: for optimization problems, generate a tree of sub-problems and find upper and lower bounds for each. The whole is more than the sum of the parts.

8. Randomized algorithms: use a random number to make a choice at some point in the algorithm. Take the correct angle and let go.