MAHARISHI UNIVERSITY OF MANAGEMENT

CS 435 – Algorithms
Discovering the Hidden Dynamics of Natural Law

Dr. Emdad Khan
(From Dr. Paul Corazza with minor updates)

June, 2015

Maharishi's Eighth Year of Invincibility

Global Raam Raj

CS 435: Algorithms
Discovering the Hidden Dynamics of the Laws of Nature

He who in action sees inaction and in
inaction sees action is wise among men.
--Bhagavad-Gita IV v18

Goals and Objectives of the Course
During the course, the student will
- Develop skill in designing algorithms and learn to represent algorithms in an implementation-neutral algorithm language
- Learn tools for evaluating efficiency of an algorithm, both empirically and analytically
- Develop the ability to demonstrate the correctness of an algorithm
- Develop discriminative ability in making the optimal selection of a data structure and/or an algorithm in a particular setting
- Learn a core of classical algorithmic solutions to both practical and theoretical problems.
- Become acquainted with techniques for improving efficiency of an algorithmic solution.
- Become acquainted with the self-referral transcendental foundation of all computation both on a theoretical and experiential level

Skills to Develop
1. Analysis of efficiency of an algorithm using
   a. big-oh etc
   b. worst-case complexity
   c. average case complexity

2. Techniques of algorithm design
   a. creating your own algorithms and express in pseudo-code
   b. using recursion
   c. using design strategies such as "divide and conquer"
   d. reducing complexity of an existing algorithm

3. Empirical tests of algorithms
   a. timing technique and test harnesses
   b. randomizing data

4. Implementation in code
   a. implement an algorithm
   b. test an algorithm
   c. problem solving: design and implement an algorithm

5. Proof of correctness of an algorithm

Class Schedule

Class is in session from 10:00 AM to 12:30 PM Mon-Sat. On Mon-Fri, the afternoon session resumes at 1:30 and continues till 3:15 (approximately). There will be a group meditation at 12:15 each day and, on Mon-Fri, at (approximately) 2:55.

Textbook

The strongly recommended text for the course is *Algorithm Design: Foundations, Analysis, and Internet Examples,* by Michael Goodrich and Roberto Tamassia, available through Amazon Books and Barnes and Noble.
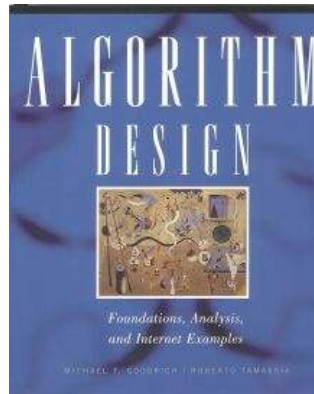
Lessons	Corresponding Sections from Textbook (and Other Resources)
Math Review	Section 1.3
1 - Introduction to Algorithms	Halting Problem: The Halting Problem Knapsack Problem: Section 5.3.3 and p. 614
2 - Introduction to Analysis of Algorithms	Sections 1.1, 1.2, 1.6. Also, p. 27.
3 - Average Case Analysis	p. 11 and Section 1.3.4
4 - More Average Case Analysis	Section 1.5
5 - Merge Sort	Section 4.1, Section 2.3.1
6 - Quick Sort	Section 4.3
7 - Lower Bound on Comparison-Based Algorithms and Radix Sort	Sections 4.4, 4.5
8 - Data Structures	Sections 2.1, 2.2, 2.5
GCD Algorithm	Section 10.1.2
9 – Binary Search Trees and Enhancements	Sections 3.1, 3.2
10 - Red-Black Trees	Section 3.3.3
11 - Heaps and Priority Queues	Section 2.4
12 - Introduction to Graph Theory	Section 6.1, Resource: Graph

	Tutor
13 - Implementing Graph Algorithms	Sections 6.1.1, 6.2, 6.3.1, 6.3.3
14 - Weighted Graphs	Sections 7.1.1, 7.3.1
15 - Hard Problems and Approximation Algorithms	Sections 13.1-13.4, Sections 5.3.1, 5.3.3

Do Labs in Groups

You will need to know how to do every homework problem yourself, but you will have the opportunity to work in a group consisting of 2-3 people. One way to do homework assignments that I do NOT recommend is to split up the assignment between group members and then combine results. This approach is almost entirely without value – better to just skip the assignment completely. The group is there for you to *discuss* homework that you have done, or to get ideas for you to work out solutions. It is not intended as a means of skipping problems.

I will assign labs every day; labs are due on the day (indicated on the calendar) on which solutions are presented to class.

Labs are submitted in paper form, in a packet, with a cover page to indicate the Lab number and group members. These submissions will be used to compute your Group Lab score.

Guidelines for Lab submission

1. Your work for a particular Lab should be submitted *as a group.*

2. Each problem in a Lab should be worked out on a *separate page* – two problems should never be worked on the same page, even if they are short.

3. If a problem asks you to write code, work it out in a development environment, but then include a hard copy of your code in your group submission, bundled with other work you have done for that problem.

4. Each hard-copy submission should have a cover page with the following information:
   - *Assignment Number*
   - *Group Name* (like "Group 1", "Group 2" etc)
   - *All names and student ID's of the members of the group*

Academic Honesty

The homework assignments provide an opportunity for you to work with the material of the course and thereby gain a deeper understanding.

Homework is *not mandatory!* It is designed to be a service to you, but if you don't want to do it, you don't need to. It is possible to get an A- even without doing any homework.

The conclusion: You should turn in only the work that you have done yourself. If you don't feel you have time for a particular assignment, then don't turn it in.

If you take any of the following alternatives instead, your assignment will receive a score of "0" and everyone in your group will be awarded the lowest possible score for professional etiquette. Moreover, there will be no opportunity to turn in a different version of the assignment to get more credit.

1. Copy a solution from another group and turn it in as your own (in this case, both groups will get a "0" for the assignment)
2. Copy a solution from a solution sheet or notes from a student that come from a previous version of this course
3. Copy a solution from the Internet

Math Quiz and Study

"Lesson 0" (in the lessons folder) serves as a math review. There is a MathReview.pdf document that reviews the main mathematical points that you should know as a pre-requisite for this course. Some of the most important points from that document are discussed in the lecture for Lesson 0, in the document MathReviewLecture.pdf. A math quiz has been placed in the lessons/Lesson 0 folder; you will submit answers to this quiz as a group (like other labs). The MathReview.pdf document will provide you with the information you will need to work through the problems on the quiz. This quiz should be done be submitted sometime before the second Friday of the course. In preparing your submission, please follow the same guidelines as for the other labs.

In-Class Exercises

From time to time, there will be in-class exercises, as time allows. These will be exercises that you will work out in your group. Groups will have the opportunity to present their solutions to these exercises, but for each problem, only one presentation will be permitted. If a group presents a correct solution in a sufficiently complete and understandable way, they will be awarded 5 Lab credits. No one is required to participate in these in-class exercises.

Exams

There will be two exams in the class. The following table lists the value and content of each exam:

Exam Number	Exam Content	Value
1	Lessons 1 – 7	46%
2	Lessons 8 - 15	46%

Grading

Your final grade will be a combination of your scores on Exams, Group Homework, and Professional Etiquette. Professional Etiquette will be evaluated with a qualitative score (Excellent, Good, Not So Good), which is an evaluation of your attendance and professional manner in class (including appearance); this score will play a role in determining how borderline grades will be decided. (Example: Your combined score for Exams, Homework, and Class Participation has put you within 1 point of an A-. If your Professional Etiquette score is Excellent, you would probably get the A-. If your Professional Etiquette score is Not So Good, you would probably get a B+.)

Evaluation Modality	Value
Combined Exam Scores	92%
Group Homework	8%
Professional Etiquette	tiebreaker

We will use the following grading scale:

Range	Letter Grade
93-100	A
90 - 92	A-
87 - 89	B+
83 - 86	B
80 - 82	B-
77 - 79	C+
73 - 76	C
67 - 72	C-
0 - 66	NC

Reference Materials

Data Structures And Algorithm Analysis In Java
    Mark Allen Weiss
    Addison-Wesley
    ISBN: 0-201-35754-2
        We have used this textbook in previous Algorithms classes.

An Introduction to Algorithms
    T.H. Cormen, C.E. Leiserson, R.L. Rivest
    McGraw-Hill, 25th printing, 2000
        The classic reference. It has everything in it. And it's inexpensive considering it spans more than 1000 pages.
    [You can download the latest version (3rd edition) free]

Data Structures, Algorithms, and Applications in Java
Sartaj Sahni
McGraw-Hill
http://www.mhhe.com/engcs/compsci/sahnijava/
Website offers Java code for many algorithms

Assignments for Algorithms

Lesson	Assignment	Suggested Reading
1	**Lab 1**	mathReview/MathReview.pdf articles/computability-theory.pdf, articles/intro-computation.pdf
2	**Lab 2**	Sections 1.1, 1.2, 1.3, 1.6 of the book. Optional: 4.1.2, 5.2 of the book.
3	**Lab 3**	Section 1.3.4 of the book
4	**Lab 4**	Section 1.5 of the book
5	**Lab 5**	Section 4.1 of the book
6	**Lab 6**	Sections 4.3, 4.7 of the book
7	**Lab 7**	Sections 4.4, 4.5, 4.6 of the book
8	**Lab 8**	Sections 2.1, 2.2, 2.5 of the book
9	**Lab 9**	Sections 2.3, 3.1, 3.2, 3.4 of the book
10	**Lab 10**	Section 3.3.3 of the book
11	**Lab 11**	Section 2.4 of the book
12	**Lab 12**	Sections 6.1, 6.2 of the book
13	**Lab 13**	Sections 6.3.1, 6.3.3 of the book
14	**Lab 14**	Sections 7.1.1 and 7.3.1 of the book
15	**Lab 15**	Sections 13.1 – 13.4 of the book

Lesson 1
Computable Functions
Creation Emerging from the Collapse of Wholeness to a Point

Wholeness of the Lesson
The reality of those mathematical functions that are used for computation in Computer Science is that they they represent only a tiny, specialized speck in the unbounded class of all number-theoretic functions. Likewise, the full range of Nature's computational power is too vast to be grasped by the intellect alone.

Main Point 1.
The computable functions that are actually used in Computer Science represent only a tiny speck of the vast range of number theoretic functions (in fact, the size of the collection of computable functions is provably smaller than the size of the set of all number theoretic functions).

Science of Consciousness: The self-referral dynamics underlying Nature's computation are unmanifest and beyond the grasp of the intellect alone.

Connecting the Parts of Knowledge
With The Wholeness of Knowledge

THE SELF-REFERRAL BASIS OF COMPUTATION

1. Computation in computer science is represented by sequential procedures.

2. The Kleene Recursion Theorem shows that every sequential procedure *f* can be *defined* by a self-referral expression of the form *f* = *F*(*f*) for some recursive operator *F: S → S*

3. *Transcendental Consciousness* is the field of pure unbounded silence, beyond the active field of Nature's computation.

4. *Impulses Within the Transcendental Field.* The hidden self-referral dynamics within the field of pure intelligence, on the ground of pure silence, give rise to the perfectly orderly unfoldment of creation.

5. *Wholeness Moving Within Itself.* In Unity Consciousness, one appreciates the flawless unfoldment of life and existence as the lively impulse of one's own pure consciousness.

Lesson 2
Introduction to Analysis of Algorithms:
Discovering the Laws Governing Nature's Computation

Wholeness of the Lesson
An algorithm is a procedure for performing a computation or deriving an output from a given set of inputs according to a specified rule. By representing algorithms in a neutral language, it is possible to determine, in mathematical terms, the efficiency of an algorithm and whether one algorithm typically performs better than another. Efficiency of computation is the earmark of Nature's self-referral performance. Contact with the home of all the laws of nature at the source of thought results in action that is maximally efficient and less prone to error.

Main Point 1.
For purposes of examining, analyzing, and comparing algorithms, a neutral algorithm language is used, independent of the particularities of programming languages, operating systems, and system hardware. Doing so makes it possible to study the inherent performance attributes of algorithms, which are present regardless of implementation details.

Science of Consciousness: This illustrates the SCI principles that more abstract levels of intelligence are more comprehensive and unifying.

Main Point 2.
Mathematical analysis is used to represent the *asymptotic behavior* of an algorithm in order to measure its efficiency. The primary concern of asymptotic analysis, both worst-case and average-case, is performance as the size (or value) of input gets large.

Science of Consciousness:
The deeper levels of an object of knowledge, which are known by deeper levels of awareness of the knower, reveal deeper truths about that object and have wider applicability than the surface values. Regular experience of transcending familiarizes awareness with its own depths, spontaneously providing deeper appreciation of all objects of perception and knowing.

Main Point 3.
An implementation consideration that can often improve the running time of an algorithm (for example, the remove duplicates algorithm) is to introduce a *tracking array* as a means to encapsulate "bookkeeping" operations, which can be done more generally using a hashtable.

Science of Consciousness: This technique is reminiscent of the Principle of the Second Element from SCI: To remove the darkness, struggling at the level of darkness is ineffective; instead, introduce a *second element* – namely, *light*. As soon as the light is introduced, the problem of darkness disappears.

Main Point 4.
Recurrence relations are used to analyze recursively defined algorithms. Just as recursion involves repeated self-calls by an algorithm, so the complexity function $T(n)$ is defined in terms of itself in a recurrence relation.

Science of Consciousness: Recursion is a reflection of the self-referral dynamics of consciousness, on the basis of which all creation emerges. Recall: "Curving back on my own nature, I create again and again." (Gita, 9.8).

<div align="center">

**Connecting the Parts of Knowledge
With The Wholeness of Knowledge**

ANALYSIS OF RECURSIVE ALGORITHMS

</div>

1. There are many techniques for analyzing the running time of a recursive algorithm. Most require special handling for special requirements (e.g. *n* not a power of 2, counting self-calls, verifying a guess).

2. The Master Formula combines all the intelligence required to handle analysis of a wide variety of recursive algorithm into a single simple formula, which, with minimal computation, produces the exact complexity class for algorithm at hand.

3. *Transcendental Consciousness* is the field beyond diversity, beyond problems, and therefore is the field of solutions.

4. *Impulses Within The Transcendental Field.* Impulses within this field naturally form the blueprint for unfoldment of the highly complex universe. This blueprint is called *Ved.*

5. *Wholeness Moving Within Itself.* In Unity Consciousness, solutions to problems arise naturally as expressions of one's own unbounded nature.

Lesson 3
Average Case Analysis
Assessing Performance through Analysis and Synthesis

Wholeness of the Lesson

Average-case analysis of performance of an algorithm provides a measure of the typical running time of an algorithm. Although computation of average-case performance generally requires deeper mathematics than computation of worst-case performance, it often provides more useful information about the algorithm, especially when worst case analysis yields exaggerated estimates. Likewise, as discussed in SCI, more successful action results from a deeper dive into silence, into pure intelligence, just as, in archery, the arrow flies truer and hits its mark more consistently if it is pulled back farther on the bow.

Main Point 1.

Average case analysis makes essential use of inherently random characteristics of the operation of an algorithm to determine an average-case asymptotic bound on its running time. Examples of such random characteristics include uniform distribution of data and random number generation. The laws underlying random behavior are used here to provide a precise estimate of the efficiency of algorithms.

Science of Consciousness: The perspective from Maharishi Vedic Science is that all expressions in the universe, however chaotic they may appear, are governed by laws of nature, which are grounded in the home of all the laws of nature, the field of pure intelligence. Accessing the home of natural law through expansion of awareness makes it possible to bring order and value into any situation, however disorderly it may appear.

**Connecting the Parts of Knowledge
With the Wholeness of Knowledge**

TYPES OF ASYMPTOTIC ANALYSIS

1. Worst-case analysis of the performance of an algorithm provides an upper bound on the running time of the algorithm.

2. Sometimes, however, worst-case analysis results in a bound that is far worse than would ever actually occur in practice. For such situations, average case analysis provides a better and more useful performance analysis, indicating the typical running time of the algorithm.

3. *Transcendental Consciousness* is the field beyond all bounds, providing the silent foundation for all computation and structure.

4. *Impulses within the Transcendental field*. All the precisely crafted structure of the universe originates from the mistake-free computational dynamics occurring within the transcendent. This unmanifest performance is called by Maharishi *Vedic Mathematics*.

5. *Wholeness moving within itself.* In Unity Consciousness, the boundaries and special characteristics that distinguish objects and individuals from one another are appreciated as lively expressions of unified wholeness.

Lesson 4
More Average Case Analysis:
Simplification by Expanding the Context

Wholeness of the Lesson
Amortized analysis provides a technique of determining average-case performance without the need to invoke the more complicated mathematical techniques that are often required in average case analysis. Amortized analysis gives a measure of the efficiency of an operation as it executes in the context of the other operations with which it is typically used.

Maharishi's Science of Consciousness: In SCI, Maharishi points out that, attempting to know the parts of knowledge separately without the wholeness of knowledge necessarily results in incomplete knowledge. Knowledge of the parts in the context of the whole results is complete knowledge.

Main Point 1.
A sorting algorithm is *inversion-bound* if it requires, on any given input array, at least as many comparisons as there are inversions in the array. Inversion-bound sorting algorithms always have an asymptotic running time that is $\Omega(n^2)$. Selection Sort, Insertion Sort and Bubble Sort are examples of inversion-bound sorting algorithms.

Science of Consciousness: Maharishi explains in SCI that knowledge is different in different states of consciousness. When consciousness is limited, it is simply not possible to see higher possibilities in life. When consciousness expands, more possibilities are seen; new directions can unfold; old problems can be solved in new ways.

Main Point 2.
In amortized analysis, we determine the total running time T_{total} of several algorithms or operations working together, for a total of, say, n executions, and then declare that the amortized running time of any one of the operations is T_{total}/n.

Science of Consciousness: Knowing the details of an object of knowledge in terms of its relationship to the wholeness to which it belongs gives more complete knowledge of the object than can be obtained by studying the object only in isolation, separate from the whole.

**Connecting the Parts of Knowledge
With The Wholeness of Knowledge**

INVERSION-BOUND ALGORITHMS

1. Insertion Sort is an inversion-bound algorithm that sorts by examining each successive value *x* in the input list and searches the already sorted section of the array for the proper location for *x*.

2. Library Sort is also a sorting algorithm which, like Insertion Sort, proceeds by examining each successive vaule *x* in the input list and searches the already sorted section of the array for proper placement. However, in this algorithm, spaces are created in the already sorted section in each pass, and searching the already sorted section is done using Binary Search. The result of these refinements is that Library Sort exceeds the limitations of an inversion-bound sorting algorithm and has average case running time that is O($n$log n).

3. *Transcendental Consciousness* is the field pure intelligence, the home of all knowledge, that field "by which all else is known".

4. *Impulses within the Transcendental field*. Maharishi explains that *knowledge has organizing power; pure knowledge has infinite organizing power*.

5. *Wholeness moving within itself.* In Unity Consciousness, the field of unlimited boundlessness is appreciated in each boundary of existence as its true nature, no different from one's own Self.

Lesson 5
Collapsing Infinity to a Point

Wholeness of the Lesson
Merge Sort is a Divide and Conquer sorting algorithm which, by overcoming the limitations inherent in inversion-bound sorting algorithms, is able to sort lists in $O(n\log n)$ time, even in the worst case. Likewise, orderly sequential unfoldment of the universe arises in the repeated collapse of infinity to a point (analysis) and expansion of point to infinity (synthesis).

Main Point 1.
By using a Divide and Conquer strategy, MergeSort overcomes the limitations that prevent inversion-bound sorting algorithms from performing faster than n^2. An essential characteristic of this strategy is the relationship of whole to part – wholes are successively collapsed and the collapsed values are combined to produce a new whole. This is different from the incremental approach of inversion-bound algorithms.

Science of Consciousness: We see here an application of the MVS principle of *akshara:* Creation arises in the collapse of the unbounded value of wholeness to a point.

Main Point 2.
Stability of a sorting algorithm requires maintenance of nonchange in the midst of change.

Science of Consciousness: This is an example in the world of sorting routines of the inner dynamics of outward success, as described in SCI: The more the inner quality of awareness remains established in silence, the more outer dynamism is supported for success and fulfillment.

Connecting the Parts of Knowledge
With The Wholeness of Knowledge

MERGE SORT

1. Inversion-bound sorting algorithms typically examine each successive element in the input array and perform a further step to place this element in an already sorted area. The style of sorting involves a *sequential unfoldment.*

2. MergeSort proceeds by repeatedly collapsing the wholeness of the current input array into parts and then synthesizing the parts into a sorted whole. This approach yields a much faster sorting algorithm.

3. *Transcendental Consciousness* is the field of *infinite correlation*, where "an impulse anywhere is an impulse everywhere," a field of "frictionless flow".

4. *Impulses within the Transcendental field.* Established in the transcendental field, action reaches fulfillment with minimum effort. Yoga is "skill in action" – efficiency in action, "doing less, accomplishing more", whereby little needs to be done to accomplish great goals.

5. *Wholeness moving within itself.* In Unity Consciousness, the field of action effortlessly unfolds as the play of one's own Self, one's own pure consciouness.

Lesson 6
QuickSort and QuickSelect:
Enlivening Hidden Laws of Nature to Manage Change

Wholeness of the Lesson
Quick Sort, another Divide and Conquer sorting algorithm, typically sorts lists even faster than Merge Sort because of its minimal use of temporary storage. Though it has a worst-case running time $\Theta(n^2)$, worst cases are extremely rare. Modifying the algorithm slightly yields a linear-time algorithm to solve the Selection Problem (finding the kth smallest element in a list).

Maharishi's Science of Consciousness: When awareness opens to finer impulses of thought, the result is heightened clarity of thinking; thoughts translate into successful action with less effort. This is the basis for more efficient behavior, allowing one to *do less and accomplish more.*

Main Point 1.
In average case analysis of both QuickSort and QuickSelect, it is observed that the actual number of self-calls required to complete sorting does not deviate much from the number of *good* self-calls that occur in the process. In a sense, the *good* self-calls cause the process as a *whole* to unfold as efficiently as possible, and the fact that they occur so frequently follows from laws governing random behavior.

Science of Consciousness: In a similar way, when the home of natural law is enlivened in individual awareness, Maharishi explains that it is "more difficult to make mistakes" – life is spontaneously lived in a way that does not create harm or problems.

Connecting the Parts of Knowledge
With The Wholeness of Knowledge

THE QUICK SELECT ALGORITHM

1. The naïve algorithm for finding the kth smallest element in a sequence of n elements requires $\Omega(n^2)$ steps. A clever alternative is to sort the sequence first and then return the value at position k. This approach runs in $\Omega(n\log n)$.

2. Using a Divide and Conquer strategy, QuickSelect locates the kth smallest element in $O(n)$ steps, on average.

3. *Transcendental Consciousness* is the field of all possibilities and the home of all knowledge. Contact with this field brings insight into new possibilities and opens awareness to expanded knowledge.

4. *Impulses Within the Transcendental Field*. The unmanifest foundation of the observable world is the lively self-interaction within pure consciousness. Within this field, creation emerges in the collapse of unboundedness to a point, and expansion from point to infinity, with infinite frequency.

5. *Wholeness Moving Within Itself*. In Unity Consciousness, the unmanifest dynamics at the source of creation are appreciated as impulses of one's own being.

Lesson 7
Lower Bound on Comparison-Based Algorithms:
Discovering the Range of Natural Law

Wholeness of the Lesson
Using the technique of decision trees, one establishes the following lower bound on comparison based sorting algorithm: Every comparison-based sorting algorithm has at least one worst case for which running time is $\Omega(n\log n)$. Bucket Sort and its relatives, under suitable conditions, run in linear time in the worst case, but are not comparison-based algorithms.

Maharishi's Science of Consciousness: Each level of existence has its own laws of nature. The laws of nature that operate at one level of existence may not apply to other levels of existence.

Main Point 1.
A decision-tree argument shows that comparison-based sorting algorithms can perform no better than $\Theta(n\log n)$. The ability to locate a lower bound on the performance of a class of algorithms – which, in the theory of algorithms, is a rare achievement – makes it possible to determine the "best" algorithm of a particular kind, such as the "best" comparison-based sorting algorithm.

Science of Consciousness: Procedures based on arriving at a goal in discrete steps always fall short of the highest possible level of efficiency, which requires *self-referral* dynamics. Nature operates on the basis of unseen self-referral dynamics – computing within itself without steps – to produce the effects that are seen as material creation.

Main Point 2.
BucketSort is an example of a sorting algorithm that runs in *linear time*. This is possible only because BucketSort does not rely primarily on comparisons in order to perform sorting.

Science of Consciousness: This phenomenon illustrates two points from SCI. First, to solve a problem, often the best approach is to bring a new element to the situation (in this case, bucket arrays); this is the Principle of the Second Element. The second point is that different laws of nature are applicable at different levels of creation. Deeper levels are governed by more comprehensive and unified laws of nature.

**Connecting the Parts of Knowledge
With the Wholeness of Knowledge**

TRANSCENDING THE LOWER BOUND ON COMPARISON-BASED ALGORITHMS

1. Comparison-based sorting algorithms can achieve a worst-case running time of $\Theta(n\log n)$, but can do no better.

2. Under certain conditions on the input, Bucket Sort and Radix Sort can sort in $O(n)$ steps, even in the worst case. The nlog n bound does not apply because these algorithms are not comparison-based.

3. *Transcendental Consciousness* is the field of all possibilities and of pure orderliness. Contact with this field brings to light new possibilities and spontaneous orderliness of all aspects of life.

4. *Impulses Within the Transcendental Field.* The organizing power of pure knowledge is the lively expression of the Transcendent, giving rise to all expressions of intelligence.

5. *Wholeness Moving Within Itself.* In Unity Consciousness, the organizing dynamics at the source of creation are appreciated as an expression of one's own Self.

Lesson 8
A Review of Data Structures:
Fully Developing the Container of Knowledge

Wholeness of the Lesson
An analysis of the average-case and worst-case running times of many familiar data structures (for instance, array lists, linked lists, stacks, queues, hashtables) highlights their strengths and potential weaknesses; clarifies which data structures should be used for different purposes; and points to aspects of their performance that could potentially be improved.

Maharishi's Science of Consciousness: Finer levels of intelligence are more expanded but at the same time more discriminating. For this reason, action that arises from a higher level of consciousness spontaneously computes the best path for success and fulfillment.

Main Point 1.
Array Lists provide $O(1)$ performance for lookup by index because of random access provided by the background array, but perform insertions and deletions in $\Theta(n)$ time, with extra overhead because of the need to break the underlying array into pieces. Linked lists improve the performance of insertion and deletion steps to $O(1)$ though locating the insertion point still requires $\Theta(n)$ time. On the other hand, linked lists, lacking random access, perform reads of all kinds in $O(n)$ time.

Science of Consciousness: Wholeness contains within it diverse – even contradictory – values; this opposite values that are integrated within wholeness make wholeness a field of all possibilities.

Main Point 2.
Stacks and queues achieve $O(1)$ performance of their main operations, which involve either reading / removing the top element or inserting a new element either at the top or the end. Stacks and queues achieve their high level of efficiency by concentrating on a single point of input (top of stack or end of queue) and a single point of output (top of stack or front of queue).

Science of Consciousness: Stacks and queues make use of the principle from Maharishi Vedic Science that the dynamism of creation arises in the concentration of dyamic intelligence to a point value ("collapse of infinity to a point").

Main Point 3.
Hashtables are a generalization of the concept of an array. They support (nearly) random access of table elements by looking up with a (possibly) non-integer key, and therefore their main operations have an average-case running time of $O(1)$.

Science of Consciousness: Hashtables illustrate the principle of *Do less and accomplish more* by providing extremely fast implementation of the main List operations.

Connecting the Parts of Knowledge
With The Wholeness of Knowledge

REALIZING THE IDEAL OF HASHTABLES

1. Assuming keys in a hashtable are randomly distributed, it can be shown that average case running time of the find, insert, and remove operations of a hashtable is $O(\alpha)$, where α is the load factor = m/n, where m is the number of keys in the table and n is the number of table slots.

2. Using universal hashing, it is possible to guarantee $O(\alpha)$ performance of find, insert, and remove operations regardless of the distribution of keys.

3. *Transcendental Consciousness* is the field of pure orderliness. Even a chaotic mind is capable of diving into this field and benefit immediately from the orderly influence that comes from contact with this field. Hidden problems and pockets of disorder are spontaneously neutralized through this process.

4. *Impulses Within the Transcendental Field.* The repeated collapse of infinity to a point and expansion of point to infinity with infinite frequency within the transcendental field produces the "hum" of creation, the integrated and even flow of all the forces underlying the manifest universe.

5. *Wholeness Moving Within Itself.* In Unity Consciousness, the unfoldment of creation from within the unmanifest is appreciated as the expression of one's own unbounded Self.

Lesson 9
Binary Search Trees:
Solving Problems By Engaging the Field of Solutions

Wholeness of the Lesson
Binary search trees make it possible to store data in memory while preserving a specified order in a way that cannot be achieved as efficiently using any kind of a list. However, their worst case performance, which can potentially occur more often than desirable, reduces their efficiency to that of a linked list. Techniques for reducing or eliminating this worst case performance include introduction of a *balance condition* to ensure the tree never becomes skewed (for example, AVL trees and Red-Black trees), and use of a *move-to-root* strategy (splay trees), to ensure that recently accessed nodes become more readily accessible.

Maharishi's Science of Consciousness: Balance is the expression of the *invincible* quality of pure creative intelligence, which preserves the integrity of unboundedness even as it is expressed within boundaries.

Main Point 1.
A *binary search tree* (BST) is a binary tree in which the BST Rule is satisfied:
> At each node *N*, every value in the left subtree of *N* is less than the value at *N*, and every value in the right subtree of *N* is greater than the value at *N*.

BSTs provide efficient search, insert, and remove operations on orderable data, transcending the limitations of these operations on list structures.

Science of Consciousness: A binary search tree is an example of the principle of Diving: Because the structure is right, the basic operations are accomplished with maximum efficiency.

Main Point 2.
AVL trees are binary search trees that remain balanced after insertions and deletions by preserving the AVL *balance condition.* The balance condition is: *For every node in the tree, the height of the left and right subtrees can differ by at most 1.* The balance condition is maintained, after insertions and deletions, by strategic use of single and double rotations. Worst-case running time for *insert, remove, find* is O(log *n*).

Science of Consciousness: The balance condition illustrates the principle that a boundary can serve to give fuller expression – rather than more limited expression – to creative intelligence.

Main Point 3.
Splay trees are binary search trees whose operations have amortized running time O(log *n*). This refinement of binary search trees averts the worst case running time of ordinary BSTs by implementing a lightweight move-to-root strategy after execution of every find, insert, and remove operation.

Science of Consciousness: Splay trees exemplify the wisdom contained in two key verses in the *Bhagavad Gita:*
- Be free of the three gunas, the field of relative existence; in other words, *transcend* (II.45)
- Established in Being, in Yoga, perform action (II.48).

Contacting and establishing oneself in one's source, one's actions produce the greatest success and life-supporting value.

CONNECTING THE PARTS OF KNOWLEDGE
TO THE WHOLENESS OF KNOWLEDGE

BALANCED BSTS

1. A Binary Search Tree can be used to maintain data in sorted order more efficiently than is possible using any kind of list. Average case running time for insertions and searchers is O(log *n*).

2. In a Binary Search Tree that does not incorporate procedures to maintain balance, insertions, deletions and searches all have a worst-case running time of $\Omega(n)$. By incorporating balance conditions, the worst case can be improved to O(log *n*).

3. *Transcendental Consciousness* is the field of perfect balance. All differences have Transcendental Consciousness as their common source.

4. *Impulses Within The Transcendental Field.* The sequential unfoldment that occurs within pure consciousness and that lies at the basis of creation proceeds in such a way that each new expression remains fully connected to its source. In this way, the balance between the competing emerging forces is maintained.

5. *Wholeness Moving Within Itself.* In Unity Consciousness, balance between inner and outer has reached such a state of completion that the two are recognized as alternative viewpoints of a single unified wholeness.

Lesson 10
Red-Black Trees:
The Principle of the Second Element

Wholeness of the Lesson
Red-black trees provide a solution to the problem of unacceptably slow worst case performance of binary search trees. This is accomplished by introducing a new element: nodes of the tree are colored red or black, adhering to the balance condition for red-black trees. The balance condition is maintained during insertions and deletions and doing so introduces only slight overhead.

Science of Consciousness: Red-black trees, as an example of BSTs with a balance condition, exhibit the Principle of the Second Element for solving the problem of skewed BSTs.

Main Point 1.
Red-black trees are binary search trees whose nodes are colored black or red in such a way that the balance condition for red-black trees is maintained:
- the root is colored black;
- if a node is red, its children are black;
- for each node N, the lengths of all paths starting at N and extending to a null reference are the same.

Science of Consciousness: Introducing auxiliary data (red/black color) and a corresponding balance condition is an example of the Principle of the Second Element from SCI, by which one arrives at a solution to a problem by introducing a new element outside the usual context of the problem.

Main Point 2.
Because of their balance condition, red-black trees always have height $O(\log n)$, so their primary operations all have running times that are $O(\log n)$.

Science of Consciousness: "Far away indeed from the balanced intellect is action devoid of greatness." (*Gita,* II.49) The "balanced intellect" – a state of life in perfect balance in which each area of life from most expressed to most subtle and refined is given due attention – arises from regular contact with the Self, the field of pure consciousness, and from bringing this unbounded value of awareness to the domain of daily living.

Main Point 3.
The integrity of red-black trees is preserved after tree operations (insertions and deletions) are performed by maintaining the balance condition after execution of each operation. This maintenance does not increase the cost of operations because it requires only constant time, involving local color changes, color flips, and rotations.

Science of Consciousness: The ability to maintain its fundamental character in the face of change is the expression of the *invincible* quality of pure consciousness. Pure consciousness, in giving rise to diversity, maintains its unbounded and immortal status. In society, this invincible quality is seen when a small percentage of a population engages in group practice of the TM and TM-Sidhi Programs – the inherent harmony of the society is enlivened to the extent that it "averts the birth of an enemy."

CONNECTING THE PARTS OF KNOWLEDGE
TO THE WHOLENESS OF KNOWLEDGE

BALANCED BSTS

1. A Binary Search Tree can be used to maintain data in sorted order more efficiently than is possible using any kind of list. Average case running time for insertions and searchers is O(log *n*).

2. In a Binary Search Tree that does not incorporate procedures to maintain balance, insertions, deletions and searches all have a worst-case running time of $\Omega(n)$. By incorporating balance conditions, the worst case can be improved to O(log *n*).

3. *Transcendental Consciousness* is the field of perfect balance. All differences have Transcendental Consciousness as their common source.

4. *Impulses Within The Transcendental Field.* The sequential unfoldment that occurs within pure consciousness and that lies at the basis of creation proceeds in such a way that each new expression remains fully connected to its source. In this way, the balance between the competing emerging forces is maintained.

5. *Wholeness Moving Within Itself.* In Unity Consciousness, balance between inner and outer has reached such a state of completion that the two are recognized as alternative viewpoints of a single unified wholeness.

Lesson 11
Priority Queues:
Rapid Access on the Ground Of Orderliness

Wholeness of the Lesson
Queues provide rapid insertions and removal of elements, adhering to the rule: "first in, first out." In a Priority Queue, one is allowed to remove elements in any desired order (not based on when the element was inserted). Implementing this idea using the *heap* data structure results in O(log *n*) performance of the main operations.

Science of Consciousness: Pure consciousness is the field of pure orderliness and the field of all possibilities. Experience of this field shows that, on the ground of the orderliness experienced from transcending, new possibilities arise in daily life to open new directions and solve problems.

Main Point 1.
A Priority Queue is essentially a queue in which each element has an associated numerical value called a *priority.* The basic operations of a Priority Queue are *removeMin* (which deletes the element of lowest priority, like *dequeue* in ordinary queues) and *insertItem* (which inserts an element in the correct location, like *enqueue*). The most efficient implementation of a Priority Queue is via *heaps* which support O(log *n*) worst-case running time for both operations, with minimal overhead. A heap is a binary tree in which every level is filled except possibly the bottom level, which is as full as possible from left to right. In addition, a binary heap satisfies the *heap order property:* For every node *X,* the priority of *X* is greater than or equal to the priority of its parent (if it has a parent).

Science of Consciousness: A Priority Queue gives a simple model of the principle of the Highest First: Putting attention on the highest value as the top priority results in fulfillment of all lower-priority values as well, in the proper time.

Main Point 2.
The Bottom Up Heap cosntruction significantly improves the efficiency of building a heap. Bottom Up Heap proceeds by dividing the input list into three: a root *r* and two remaining halves. These remaining halves are recursively organized into separate heaps, and then joined together to form the final heap with root *r*.

Science of Consciousness: This process provides an analogy for the process of unfoldment of creation from the unmanifest. In this process, three (rishi, devata, chhandas) emerge from one and then, through self-referral dynamics (a kind of "recursion"), unfold sequentially to form the blueprint of the universe.

Connecting the Parts of Knowledge
With the Wholeness of Knowledge:

1. HeapSort is a fast sorting algorithm that proceeds by first building a heap from an input list and then repeatedly removing the max value to form final sorted output. Each step requires nlog n steps, asymptotically.

2. The Bottom Up Heap construction makes it possible to perform the first phase of HeapSort – the heap-building phase – in $O(n)$ time. This is accomplished by dividing the input list into three, recursively building heaps in two of the three parts, and joining the parts together, returning the final heap.

---

3. *Transcendental Consciousness* is the unmanifest field of pure consciousness, having a 3-in-1 structure, consisting of rishi, devata, and chhandas.

4. *Impulses within the Transcendental field*: Within the transcendental field, its fundamental 3-in-1 structure gives rise to a self-referral unfoldment of the blueprint of creation, the *Ved.*

5. *Wholeness moving within itself:* In Unity Consciousness, every aspect of experience is appreciated in terms of the 3-in-1 dynamics of consciousness.
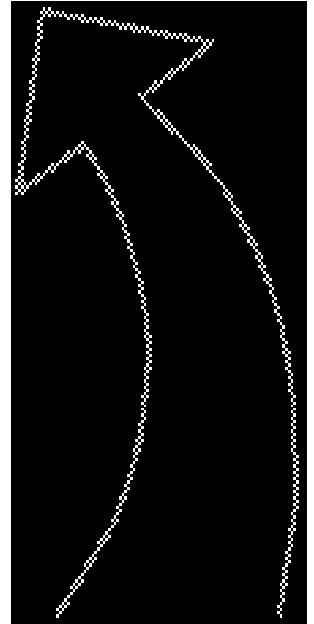
Lesson 12
Graphs:
Combinatorics of Pure Intelligence

Wholeness of the Lesson

Graphs are data structures that do more than simply store and organize data; they are used to model interactions in the world. This makes it possible to make use of the extensive mathematical knowledge from the theory of graphs in solving problems abstractly, at the level of the model, resulting in a solution to the real-world problem.

Science of Consciousness: "Knowledge is structured in consciousness" – by contacting the "home of all knowledge" at the source of thought, one spontaneously engages Nature's intelligence, Nature's know-how, in one's daily activities. Real-world problems are solved on the basis of the abstract dynamics of intelligence experienced effortlessly during meditation and lived effortlessly in daily life.

Main Point 1.

A graph consists of vertices and edges, which may all be undirected (producing an *undirected graph*) or all directed (producing a *directed graph*). Virtually all finite patterns in the universe can be modeled using graphs since they can be seen abstractly as objects and their relationships.

Science of Consciousness: Directed graphs provide an especially good analogy for the unfoldment of creation according to Maharishi Vedic Science. In such a graph, when vertices v_1, v_2 are joined by a directed edge e, $v_1 \to v_2$, v_1 can be seen to act upon or "know" v_2. Here, v_1 plays the role of rishi, v_2 that of chhandas, and the edge e between them, devata. As with directed graphs, everything in the universe, according to Maharishi Vedic Science, arises from the interacting dynamics of rishi, devata, and chhandas.

Main Point 2.

A *graph algorithm* is a procedure performed on a graph to discover information about its structure or to transform it in some way. For example, graph algorithms can be used to determine whether a graph is complete, bipartite, acyclic, and/or connected. No efficient algorithm is known for determining whether a graph has certain characteristics, such as that of being Hamiltonian, or whether it has a vertex cover of a specified size.

Science of Consciousness: When the means of gaining knowledge is restricted to the objective approach based on the intellect, knowledge must remain incomplete and ultimately unsatisfying. When the highest value of the subject of knowledge is included in the process of gaining knowledge, that knowledge is both complete and self-validating.

CONNECTING THE PARTS OF KNOWLEDGE
TO THE WHOLENESS OF KNOWLEDGE

GRAPH PROBLEMS BELONGING TO NP

1. Some problems about graphs are not known to have algorithmic solutions that run in polynomial time – these are known as *hard problems.*

2. Some hard graph problems are tractable enough to make it possible to verify a solution in polynomial time, such as the Hamiltonian Graph and Vertex Cover problems. These problems belong to the class *NP.*

3. *Transcendental Consciousness* is the home of all knowledge. Contact with this field makes all problems tractable.

4. *Impulses Within The Transcendental Field.* The structure of knowledge, the foundation of all things known, of all answers to all questions, of all solutions to all problems, arises in the simple interaction of pure consciousness with itself. Contact with this field saturates the individual life with the lively potential for insight, success, and a problem-free life.

5. *Wholeness Moving Within Itself.* In Unity Consciousness, problems and their solutions are seen to arise as impulses of one's own being. Problems are intimate, and so are their solutions.

Lesson 13
Implementing Graphs and Graph Algorithms:
Knowledge Has Organizing Power

Wholeness of the Lesson
Much information about the structure of a given graph – information that is useful to know for applications – can be obtained from a variety of efficient graph algorithms. DFS and BFS are foundational or "template" algorithms that can be tailored to compute such things as the connected components of a graph and to determine whether the graph contains a cycle and whether it is bipartite. Foundational operations on a graph can be seen as the basic operations of Graphs as an abstract data type; the Graph data type can be implemented as a software object, consisting of a set of vertex objects and a set of edge objects, together with implementations of fundamental operations.

Science of Consciousness: Implementation of the Graph data type as software objects reflects the Maharishi Vedic Science theme of unfoldment of creation through interaction of rishi, devata and chhandas. Here, the Graph data type corresponds to rishi; the graph software object corresponds to chhandas; and the realization of structural and dynamic elements of a Graph as elements of implementation corresponds to devata.

Main Point 1.
The BFS and DFS algorithms are procedures for visiting every vertex in a graph. BFS proceeds "horizontally", examining every vertex adjacent to the current vertex before going deeper into the graph. DFS proceeds "vertically", following the deepest possible path from the starting point and, after reaching the end, backtracks to follow another path to the end starting from some earlier point on the first path, and continues in this way till all vertices have been reached.

Science of Consciousness: These approaches to graph traversal are analogous to the horizontal and vertical means of gaining knowledge, as described in SCI: The horizontal approach focuses on a breadth of connections at a more superficial level, and reaches deeper levels of knowledge more slowly. The vertical approach dives deeply to the source right from the beginning; having fathomed the depths, subsequent gain of knowledge in the horizontal direction enjoys the influence of the depths of knowledge already gained.

Main Point 2.
To answer questions about the structure of a graph *G*, such as whether it is connected, whether there is a path between two given vertices, and whether the graph contains a cycle, it is sufficient to use DFS to compute the connected components of the graph. Based on this one piece of information, all such questions can be answered efficiently.

Science of Consciousness: This phenomenon illustrates the SCI principle of the *Highest First*: Experience the home of all knowledge first, and all particular expressions of knowledge become easily accessible.

Connecting the Parts of Knowledge
With the Wholeness of Knowledge:

DETECTING BIPARTITE GRAPHS

1. The BFS algorithm provides an efficient procedure for traversing all vertices in a given graph.

2. By tracking edges and levels during execution of the BFS algorithm, it is possible to detect the presence of an odd cycle (this occurs if an already visited vertex is found which is at the same level as the current vertex and is also adjacent to it). This allows us to determine whether the graph is bipartite.

3. *Transcendental Consciousness* is the field of all possibilities, located at the source of thought by an effortless procedure of transcending.

4. *Impulses within the Transcendental field*: The entire structure of the universe is designed in seed form within the transcendental field, all in an effortless manner.

5. *Wholeness moving within itself*: In Unity Consciousness, each expression of the universe is seen as the effortless creation of one's own unbounded nature.

Lesson 14
Algorithms For Weighted Graphs:
Creative Intelligence Manifesting As Material Creation

Wholeness of the Lesson
Weighted graphs are (undirected) graphs that have weights associated with each edge. They provide useful models for a wide variety of real-world systems, including transportation networks, electronic circuits, and computer networks. Two questions that typically need to be answered when working with such weighted graphs are (1) What is the least costly path between two given vertices of the graph? (2) What is the least costly subgraph of the given graph which includes all the vertices of the given graph? Dijkstra's Shortest Path Algorithm provides an efficient solution to the first question; Kruskal's Minimum Spanning Tree Algorithm provides an efficient solution to the second.

Science of Consciousness: Solutions to optimization problems of all kinds give expression to Nature's tendency to achieve the most possible with the least expenditure of energy. Enlivening the home of all the laws of nature through contact with transcendental consciousness spontaneously results in action and behavior that achieve their aim most efficiently.

Main Point 1.
Dijkstra's algorithm is an example of a *shortest-path algorithm* – an algorithm that efficiently ($O(m\log n)$) computes the shortest distance between a given vertex and other vertices in a graph.

Science of Consciousness: Analogously, Nature itself is known to obey the law of least action – Nature does the least possible amount of work to proceed from one location or state to another. Nature's way of achieving this makes use of computational dynamics that involve "no effort" and no steps.

Main Point 2.
Kruskal's algorithm efficiently ($O(m\log n)$) computes a *minimum spanning tree* in a weighted graph. It proceeds by choosing edges with minimum possible weight subject to the constraint that its endpoints live in distinct clusters. This implementation of a "greedy" strategy is successful because, in this case, choosing the edge that appears to be optimal at each step of the algorithm accurately reflects the global reality concerning optimal edge selection.

Science of Consciousness: An enlightened person maintains unbounded awareness along with localized awareness. The behaviors of such a person are globally optimal for all problems.

CONNECTING THE PARTS OF KNOWLEDGE
WITH THE WHOLENESS OF KNOWLEDGE

COMPUTING MINIMUM SPANNING TREES

1. A Minimum Spanning Tree can be obtained from a weighted graph $G = (V,E)$ by examining all possible subgraphs of G, and extracting from those that are trees the tree having the smallest sum of edge weights. This procedure runs in $\Omega(2^n)$, where n is the number of vertices of the graph.

2. Kruskal's Algorithm is a highly efficient procedure ($O(m\log n)$) for finding a minimum spanning tree in a graph G. It proceeds by choosing edges with minimum possible weight subject to the constraint that its endpoints live in distinct clusters.

3. *Transcendental Consciousness*, the simplest form of awareness, is the source of effortless right action.

4. *Impulses Within the Transcendental Field.* Effortless, economical, mistake-free creation arises from the self-referral dynamics of the field of pure consciousness.

5. *Wholeness Moving Within Itself.* In Unity Consciousness, optimal solutions arise as an effortless unfoldment within one's unbounded nature.

Lesson 15
NP-Complete Problems:
Handling Problems from the Field of All Possibilities

Wholeness of the Lesson
Decision problems that have no known polynomial time solution are considered *hard*, but hard problems can be further classified to determine their degree of hardness. A decision problem belongs to NP if there is a polynomial p and an algorithm A such that for any instance of the problem of size n, a correct solution to the problem can be *verified* using A in at most $p(n)$ steps. In addition, the problem is said to be *NP-complete* if it belongs to NP and every NP problem can be polynomial-reduced to it.

Science of Consciousness: The simplest form of awareness, experienced at the source of thought during TM practice, transcends the intellect. The Bhagavad Gita declares "beyond even the intellect is he'' (III.42). But even the course of action in the manifest field of existence, including the subtle effects of every thought and action on the span of the universe, is just too complex to be grasped by the intellect alone. The Gita declares "unfathomable is the course of action" (IV.17).

Main Point 1.
The hardest NP problems are *NP-complete.* These require the highest degree of creativity to solve. However, if a polynomial-time algorithm is found for any one of them, then all NP problems will automatically be solved in polynomial time.

Science of Consciousness: This phenomenon illustrates the fact that the field of pure consciousness, the source of creativity, is itself a field of *infinite correlation* – "an impulse anywhere is an impulse everywhere".

Main Point 2.
NP-completeness of a problem can be established by polynomially reducing another NP-complete problem to it. Moreover, the statement P = NP is true if and only if a polynomial time algorithm can be found for a *single* NP-complete problem.

Science of Consciousness: All problems at all levels of life are solved by solving one fundamental problem: the problem of *ignorance* of one's essential nature. This is the Principle of the Highest First: capture the fort of pure consciousness first, and all other values in life will fall into place automatically.

Main Point 3.
Infeasibility of known solutions to NP problems can be addressed in several ways:
(1) Find a more efficient algorithm (example: use the technique of *dynamic programming*).
(2) Use an *approximation algorithm* which efficiently produces near-optimal solutions.
(3) Find a *probabilistic algorithm* which efficiently outputs answers that are correct with "high probability".
(4) Make use of the hardness of a problem as a central element in a security protocol.

Science of Consciousness: Pure consciousness is a field of infinite creativity and all possibilities. Contact with this field leads to a life that handles problems creatively and even turns a potential enemy into a friend.

Connecting the Parts of Knowledge
With the Wholeness of Knowledge

NP-COMPLETE PROBLEMS

1. There are many natural decision problems in Computer Science for which feasible solutions are needed, but which are NP-complete. Therefore, there is little hope of finding such solutions.

2. The hardness of certain NP-complete problems can be used to ensure the security of certain cryptographic systems.

3. *Transcendental Consciousness* is a field of all possibilities and infinite creativity.

4. *Impulses Within the Transcendental Field.* Pure consciousness, as it prepares to manifest, is a "wide angle lens" making use of every possibility for creative ends.

5. *Wholeness Moving Within Itself.* In Unity Consciousness, awareness does not get stuck in problems; problems are seen as steps of progress in the unfoldment of the dynamics of consciousness.