MAHARISHI UNIVERSITY OF MANAGEMENT

CS 435 – Algorithms
Discovering the Hidden Dynamics of Natural Law

Instructor
Mei Li

2018

Maharishi's eleventh Year of Invincibility

Global Raam Raj

			Algorithms			
Week	**Monday**	**Tuesday**	**Wednesday**	**Thursday**	**Friday**	**Saturday**
1	**AM: Lesson 1**: *Introduction to Algorithms* **PM: Lab 1** *Math Review*	**AM: Lesson 2**: *Intro to Analysis Of Algorithms* **PM: Lab 2**	**AM: Lesson 2**: (continued) **PM: Lab 2** **Lab 1 is due**	**AM: Lesson 3**: *Average Case Analysis* **PM: Lab3**	**AM: Lesson 4**: *MergeSort* **PM: Lab 4** **Lab 2 is due**	**AM: Lesson 5**: *QuickSort* **Weekend: Lab5** **Lab 3 is due**
2	**AM: Lesson 6**: *Lower Bound on Comparison-Based Algorithms And RadixSort* **PM: Lab 6** **Lab 4 is due**	**AM: Lesson 7**: *Review of Data Structures* **PM: Lab 7** **Lab 5 is due**	**AM: Lesson 8**: *Red-Black Trees* **PM: Lab 8** **Lab 6 is due**	**AM: Lesson 9**: *Heaps and Priroity Queues* **PM: Lab 9** **Lab 7 is due**	**AM:** *Review for Midterm* **PM: Study for midterm** **Labs 8, 9 are due**	*Midterm Exam*
3	**AM: Lesson 10**: *Algorithm Design* **PM: Lab10**	**AM: Lesson 10**: (continued) **PM: Lab 10**	**AM: Lesson 11**: *Graphs and Graph Traversal* **PM: Lab 11**	**AM: Lesson 12**: *Directed Graphs* **PM: Lab 12** **Lab 10 is due**	**AM: Lesson 13**: *Weighted Graphs – Shortest Path Algorithms* **PM: Lab 13** **Lab 11 is due**	**AM: Lesson 13**: *Weighted Graphs – Minimum Spanning Tree Algorithms* **PM: Lab 13** **Lab 12 is due**
4	**AM: Lesson 14**: *Hard Problems* **PM: Lab 14** **Lab 13 is due**	**AM: Lesson 14**: *Hard Problems* **PM: Lab 14** **Review lab solutions**	**AM**: Review for Final exam **PM: Study for final** **Lab 14 is due**	*Final Exam*		

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.
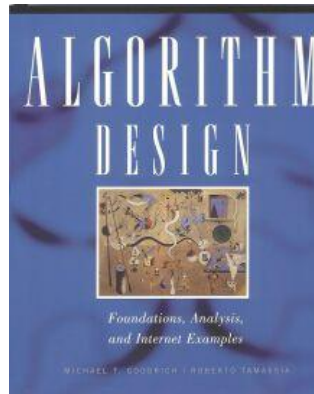
Lessons	Corresponding Sections from Textbook (and Other Resources)
Math Review	Section 1.3
1 - Introduction to Algorithms	Halting Problem: The Halting Problem Knapsack Problem: Section 5.3.3 and p. 614
2 - Introduction to Analysis of Algorithms	Sections 1.1, 1.2, 1.6. Also, p. 27.
3 - Average Case Analysis	p. 11 and Section 1.3.4, 1.5
4 - Merge Sort	Section 4.1, Section 2.3.1
5 - Quick Sort	Section 4.3
6 - Lower Bound on Comparison-Based Algorithms and Radix Sort	Sections 4.4, 4.5
7 - Data Structures	Sections 2.1, 2.2, 2.5, 3.1, 3.2
GCD Algorithm	Section 10.1.2
8 - Red-Black Trees	Section 3.3.3
9 - Heaps and Priority Queues	Section 2.4
10 - Algorithm Design	Sections 5.1, 5.2, 5.3
11 - Introduction to Graph Theory	Section 6.1, Resource: Graph Tutor

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

Labs

I would encourage you to work in groups since some labs may seem long… But everyone needs to submit labs <u>individually</u>.

Labs are submitted in sakai. Submit your work in either Word or pdf format.

If a problem asks you to write code, work it out in a development environment, but then copy your code to the word document.

Academic Honesty

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

You should turn in only the work that you have done yourself. If you take any of the following alternatives instead, your assignment will receive a score of "0".

1. Copy a solution from another student and turn it in as your own.
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

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 – 9	45%
2	Lessons 10 - 14	45%

Grading

Your final grade will be a combination of your scores on Exams, Homework, Professional Etiquette and Morning Meditation. Professional Etiquette is an evaluation of your attendance and professional manner in class. You may earn extra credit towards your final grade if you have outstanding attendance at the morning meditations this block.

> 70% and above: .5% EC (16 days in a standard block)
> 80% and above: 1% EC (18 days in a standard block)
> 90% and above: 1.5% EC (20 days in a standard block)

Evaluation Modality	Value
Combined Exam Scores	90%
Homework	8%
Professional Etiquette	2%
Morning Meditation Bonus	0%-1.5%

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.

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

Lesson 1
Solving Problems with Algorithms
Creation Emerging from the Collapse of Wholeness to a Point

Wholeness of the Lesson

Much of the power of modern software arises from the fact that it makes use of highly efficient algorithmic solutions to a variety of problems. These solutions represent the concentrated intelligence of great thinkers of the past half-century. Yet the range of problems for which such solutions are known is actually only a tiny speck in the landscape of abstract problems. Likewise, the full range of Nature's computational power is too vast to be grasped by the intellect alone, yet harnessing even a little of that power is enough to produce powerful results.

Main Point 1.

It has turned out that the problems that are most needed to be solved for the purpose of developing modern-day software projects happen to lie in the very specialized class of P Problems – problems that have feasible solutions. For these problems, the creativity and intelligence of the industry has been concentrated to a point; the algorithms that have been developed for these are extremely fast and highly optimized. At the same time, this tiny point value reveals how vast is the range of problems that cannot be solved in a feasible way. These points illustrate that creative expression arises in the collapse of unboundedness to a point, and also that unboundedness itself is beyond the grasp of the intellect.

Connecting the Parts of Knowledge
With The Wholeness of Knowledge

THE SELF-REFERRAL BASIS OF COMPUTATION

1. The functions that are used and analyzed in practice in the development and analysis of software are the *polynomial bounded functions*.

2. The polynomial bounded functions form only a tiny speck in the expansive class of all number-theoretic functions N → N.

3. *Transcendental Consciousness* is the fully expanded field of consciousness, beyond even the most general notion of "function".

4. *Impulses Within The Transcendental Field*. As pure consciousness becomes conscious of itself, it undergoes transformational dynamics, as knower, known, and process of knowing interact. The process of knowing is the first sprout of the notion of "function."

5. *Wholeness Moving Within Itself*. In Unity Consciousness, the transformational dynamics of consciousness, at the basis of all of creation, are appreciated as the lively impulses of one's own consciousness, one's own being.

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).

Connecting the Parts of Knowledge
With The Wholeness of Knowledge

ANALYSIS OF RECURSIVE ALGORITHMS

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.

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 $\Theta(n^2)$. Selection Sort, Insertion Sort and Bubble Sort are examples of inversion-bound sorting algorithms.

Maharishi explains in SCI that knowledge is different in different states of consciousness. When consciousness is bounded, 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.

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

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(nlog n).

3. *Transcendental Consciousness* is the field pure intelligence, the home of all knoweldge, 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 4
Merge Sort:
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 5
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 *k*th 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 6
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 $n \log n$ 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 7
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, binary search trees) 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. Likewise, 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.

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. 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: Wholeness contains within it diverse – even contradictory – values; these opposite values that are integrated within wholeness make wholeness a field of all possibilities. Experience of this wholeness value brings into our lives the ability to handle and make good use of opposite values.

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 2.

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

Main Point 3.

AVL trees are binary search trees in which 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*). The balance condition illustrates the principle that boundaries can serve to *give expression to* boundless intelligence rather than simply *limiting* that intelligence.

三十幅共一轂: 當其無有車之用。 埏埴以為器, 當其無有器之用。 鑿戶牖以為室, 當其無有室之用。 故有之以為利, 無之以為用。	Thirty spokes share the wheel's hub; It is the center hole that makes it useful. Shape clay into a vessel; It is the space within that makes it useful Cut doors and windows for a room; It is the holes which make it useful. Therefore, profit comes from what is there; Usefulness comes from what is not there. *Daodejing 11*

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 8
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 9
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:

BOTTOM UP HEAP CONSTRUCTION

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 10
Algorithm Design:
Structuring the Laws of Nature in Individual Awareness

Wholeness of the Lesson

Algorithm Design is an intelligent approach to solving problems with algorithms. Rather than simply trying to tackle a problem haphazardly, one can determine whether the problem has the characteristics that make it easy to solve using one of many known algorithm design strategies.

Maharishi's Science of Consciousness: The textbook of SCI, the Bhagavad Gita, declares "Yogastah Kuru Karmani" – Established in Being, perform action. When awareness has a chance to be bathed in the field of pure orderliness, activity afterwards has an orderly quality that naturally leads to success and achievement.

Main Point 1.

The Greedy algorithm design attempts to solve a problem by accepting, at each step of computation, a value that is optimal at that step, without regard for future steps or for the emerging context. The greedy strategy is "doing without planning for the future." In life, the greedy strategy, according to SCI, works well only when individual life is directed by a higher quality of intelligence. Then the "planning" is done by cosmic intelligence. But until the home of all the laws of nature is established in individual awareness, it is better to plan carefully for the future and to avoid the "greedy strategy."

Main Point 2.

The Divide and Conquer algorithm design attempts to solve a problem by breaking it into small disjoint subproblems, solving the subproblems separately, and combining into a final solution. This pattern of solving problems is the pattern outlined in the ancient texts by which structure emerges from the unmanifest level of existence: Analysis into parts, synthesis of parts into whole.

Main Point 3.

Dynamic Programming is an algorithm design technique that arrives at an optimal solution by computing optimal solutions to overlapping subproblems, storing the results (memoization) to avoid redundant computations, and then combining subproblem solutions to obtain the final solution. In SCI, it is observed that to restore completeness in the life of the individual – to solve the problem of life as a human being – we must restore the *memory* of our unbounded nature. For that, we repeatedly open awareness to its unbounded nature, through the process of transcending.

CONNECTING THE PARTS OF KNOWLEDGE
TO THE WHOLENESS OF KNOWLEDGE

1. Dynamic programming can transform an infeasible (exponential) computation into one that can be done efficiently.

2. Dynamic programming is applicable when many subproblems of a recursive algorithm overlap and have to be repeatedly computed. The algorithm stores solutions to subproblems so they can be retrieved later rather than having to re-compute them.

3. Transcendental Consciousness is the silent, unbounded home of all the laws of nature.

4. *Impulses within Transcendental Consciousness:* The dynamic natural laws within this unbounded field are perfectly efficient when governing the activities of the universe.

5. *Wholeness moving within itself:* In Unity Consciousness, one experiences the laws of nature and all activities of the universe as waves of one's own unbounded pure consciousness.

Lesson 11
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: Our own deeper levels of intelligence exhibit more of the characteristics of Nature's intelligence than our own surface level of thinking. Bringing awareness to these deeper levels, as the mind dives inward, engages Nature's intelligence, Nature's know-how, and this value is brought into daily activity. The benefit is greater ability to solve real-world problems, meet challenges, and find the right path for success.

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, $v_1 \rightarrow v_2$, 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 12
Implementing Graphs and Graph Algorithms:
Knowledge Has Organizing Power

Wholeness of the Lesson
The body of knowledge in the field of Graph Theory becomes accessible in a practical way through the Graph Abstract Data Type, which specifies the computations that a Graph software object should support. The Graph Data Type is analogous to the human physiology: The abstract intelligence that underlies life relies on the concrete physiology to find expression in thinking, feeling, and behavior in the physical world.

Main Point 1.
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. 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.

Main Point 2.
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 till all vertices have been reached. 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.

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 13
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 14
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.