CSC 213

INTRODUCTION TO ALGORITHMS AND APPLICATIONS

Course Summary By Paul(Dr Deja Vu)

Algorithm

An algorithm can be defined as any well stated computational procedure that takes place in a set of values as input and produces a set of values as output.

It is therefore a set of computational steps that transforms input into output and it is used to solve problems.

Algorithmics

It can be defined as the science that allows designers to study and evaluate the effects of algorithm based on various factors such that the best algorithm is chosen to satisfy a particular circumstance in a way that satisfies the need of the algorithm.

Program

A Program Is an expression of an algorithm in a programming language.

Reasons For Analysis

1. Writing a working program is not good enough.

2. The program may be inefficient.

3. If you run your program on a large data set, the running time becomes an issue of concern.

Types Of Algorithm

1. Probabilistic Algorithm

2. Heuristic Algorithm

3. Approximate Algorithm

Probabilistic Algorithm

This allows values to be used in such a way that the values are used as subject of probability operation which occurs by chance.

Heuristic Algorithm

This is the type that is based largely on optimism and often with minimal theoretical support.

Approximate Algorithm

In this type answer is obtained that is as precise to the required answer in approximate decimal notation.

Properties Of An Algorithm

1. Finiteness: An algorithm should end in finite amount of steps.

2. Definiteness: Each Step or instruction must be clear.

3. Effectiveness: Each Step must perform it's step effectively.

4. Correctness: For every input there must be a corresponding output.

5. It must have an input.

6. It must have an output.

Pseudo code

This is a mixture of a natural language and programming language like constructs. A pseudo code is usually more precise than a natural language and it's usage often yields more successful algorithm descriptions.

Flowchart

This is a method of expressing an algorithm by a collection of connected geometric shapes containing descriptions of algorithms steps.

Algorithm Correctness

This is to ascertain that the algorithm yields a required result for every legitimate input in a finite amount of time.

Example: Correctness of Euclid's algorithm for computing the greatest common divisor stems from correctness of the equality.

gcd(m, n) = gcd(n, m mod n)

Efficiency of An Algorithm

This can be defined and investigated with mathematical rigor. Two kinds of algorithm efficiency are:

Time Efficiency: This indicates how fast the algorithm runs.

Space Efficiency: This indicates how much memory the memory needs.

Sorting Problem

This requires the rearrangement of data items of a given list in an ascending or descending order.

Worst Case Efficiency

This is the efficiency for the worst case input of size n, which is an input(or inputs) of size n for which the algorithm runs the longest among all possible inputs of that size.

EX: if you want to sort a list of numbers in ascending order when the numbers are given in descending order. In this running time will be the longest.

Best Case Efficiency

This is the efficiency for the best case input of size n, which is an input of size n for which the algorithm runs the shortest among all possible inputs of that size.

Worst Case Time Complexity

This is the maximum time required for program execution.

Best Case Time Complexity

This is the Minimum time required for program execution.

Average Case Time Complexity

This is the Average time required for program execution.

Order Of Growth

An **order of growth** is a set of functions whose asymptotic **growth** behavior is considered equivalent.

For example, 2n, 100n and n + 1 belong to the same **order of growth**, which is written O(n) in Big-Oh notation and often called linear because every function in the set grows linearly with n.

Divide and Conquer

This is the best known general algorithm design technique. It works according to the following general plan.

I. A problem's instance is divided into several smaller instances of the same problem, ideally of about the same size.

ii. The smaller instances are solved.

III. if necessary, the solutions obtained for the smaller instances are combined to get a solution to the original problem.

Recurrence

A recurrence is an equation or inequality that describes a function in terms of its value on smaller inputs.

Methods Of Solving Recurrence

1. Substitution Method: We make a guess for the solution and then we use mathematical induction to prove the guess is correct or incorrect.

2. **Recurrence Tree Method:** In this method, we draw a recurrence tree and calculate the time taken by every level of tree. Finally, we sum the work done at all levels.

3. Master Theorem: This provides an asymptotic analysis for recurrence relations of types that occur in the analysis of many divide and conquer algorithms.

P Class

The complexity class **P** is the set of all decision **problems** that can be solved with worst-case polynomial time-complexity.

NP Problem

It is the class of decision problem that can be solved by non-deterministic polynomial algorithms. Most decision problems are in NP.

First of all, this class includes all the problem in P. This class of problems is called Non-deterministic polynomial.

NP Complete Problem

A decision problem is said to be NP-complete if

1. It belongs to class NP

2. Every Problem in NP is polynomial reducible to D.

NP Hard Problem

The motion of an NP-Hard, problem can be divided more formally by extending the notion of reducibility to problems

that are not necessary in class NP including optimization problems.

Complexity Of An Algorithm

Complexity of an **algorithm** is a measure of the amount of time and/or space required by an **algorithm** for an input of a given size (n).

When is a decision problem said to be polynomial reducible?

This is said when a decision problem D2 if their exist a function that transfers instances of D1 to instances D2 such that

I. it maps all yes instances of D1 to yes instances of D2 and all no instances of D1 to no instances of D2.

ii. it is computable by a polynomial time algorithm.

Big Oh Notation, O

The notation O(n) measures the worst case time complexity or the longest amount of time an algorithm can possibly take to complete.

Omega Notation, Ω

The notation Ω(n) measures the best case time complexity or the best amount of time an algorithm can possibly take to complete.

Theta Notation, θ

The notation θ(n) is the formal way to express both the lower bound and the upper bound of an algorithm's running time.

Small Omega Notation

This is a rough estimate of the order of the growth whereas Big **Omega** (Ω) may represent exact order of growth.

Little Omega

Little Omega (ω) is a rough estimate of the order of the growth whereas Big **Omega** (Ω) may represent exact order of growth.

Binary Tree

This is a tree type non linear data structure with a maximum of two children for each parent.

Binary Search

This is a search algorithm that finds the position of a target value within a sorted array. Binary search compares the target value to the middle element of the array.

Sequential Algorithm

A **sequential algorithm** or serial **algorithm** is an **algorithm** that is executed **sequentially** – once through, from start to finish, without other processing executing – as opposed to concurrently or in parallel.

Parallel Algorithm

A **parallel algorithm** is an **algorithm** that can execute several instructions simultaneously on different processing devices and then combine all the individual outputs to produce the final result.

Exact Algorithm

Exact algorithms are **algorithms** that always find the optimal solution to a given optimization problem.

Algorithm Design Strategies

1. Divide and Conquer Approach: It is a top-down approach. The algorithms which follow the divide & conquer techniques involve three steps:

- Divide the original problem into a set of sub problems.
- Solve every sub problem individually, recursively.
- Combine the solution of the sub problems (top level) into a solution of the whole original problem.

2. Greedy Technique: Greedy method is used to solve the optimization problem. An optimization problem is one in which we are given a set of input values, which are required either to be maximized or minimized (known as objective), i.e. some constraints.

3. **Dynamic Programming:** Dynamic Programming is a bottom-up approach we solve all possible small problems and then combine them to obtain solutions for bigger problems.

4. **Branch and Bound:** In Branch & Bound algorithm a given sub problem, which cannot be bounded, has to be divided into at least two new restricted sub problems.

5. **Brute Force:** Brute Force says that for any given problem, try out all possible solution and pick up the desired solution.

6. **Backtracking Algorithm:** Backtracking Algorithm tries each possibility until they find the right one. It is a depth-first search of the set of possible solution.

7. **Randomized Algorithm:** A randomized algorithm uses a random number at least once during the computation make a decision.

Amortized Efficiency.

Amortized analysis is a method for analyzing a given algorithm's complexity, or how much of a resource, especially time or memory, it takes to execute.

Polynomial Algorithm

The polynomial-time algorithm is an **algorithm** whose execution **time** is either given by **a polynomial** on the size of the input, or can be bounded by such a **polynomial**.

Exponential Algorithm

An **algorithm** is said to be **exponential time**, if T(n) is upper bounded by 2, where poly(n) is some polynomial in n.

Symbols and rules for drawing a good flowchart

1. Use Consistent Design Elements. Shapes, lines and texts within a flowchart diagram should be consistent.

2. Keep Everything on One Page.

3. Flow Data from Left to Right.

4. Use a Split Path Instead of a Traditional Decision Symbol.

5. Place Return Lines Under the Flow Diagram.

Steps of Development Of An Algorithm

- **Step** 1: Obtain a description of the problem. This **step** is much more difficult than it appears....
- **Step** 2: Analyze the problem.
- **Step** 3: **Develop** a high-level **algorithm**.
- **Step** 4: Refine the **algorithm** by adding more detail.
- **Step** 5: Review the **algorithm**.

Insertion Sort

Insertion sort is a simple **sorting** algorithm that builds the final **sorted** array (or list) one item at a time.

Merge Sort

Merge sort is a sorting technique based on divide and conquer technique. With worst-case time complexity being O(n log n), it is one of the most respected algorithms. Merge sort first divides the array into equal halves and then combines them in a sorted manner.

Bubble Sort

Bubble sort, sometimes referred to as sinking sort, is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements and swaps them if they are in the wrong order.

Radix Sort

Radix sort is an integer **sorting** algorithm that **sorts** data with integer keys by grouping the keys by individual digits that share the same significant position and value (place value).

Selection Sort

This **sorting** algorithm is an in-place comparison-based algorithm in which the list is divided into two parts, the **sorted** part at the left end and the unsorted part at the right end.

Quick Sort

In Quick Sort, we select an element as pivot, partition the **array** around pivot and recur for subarrays on left and right of pivot.

Backtracking

Backtracking is an algorithmic-technique for solving problems recursively by trying to build a solution incrementally, one piece at a time, removing those solutions that fail to satisfy the constraints of the problem at any point of time.

Primitive Operations in the time Complexity of a program.

1. Evaluating an expression.
2. Assigning a value to a variable.
3. Indexing into an array.
4. Calling a method.
5. Returning from a method.

Greedy Algorithm

A **greedy algorithm** is a simple, intuitive **algorithm** that is used in optimization problems.

Knapsack Problem

The **knapsack problem** is a **problem** in combinatorial optimization: Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.

Hamiltonian Tour

Hamiltonian path is a path in an undirected or directed graph that visits each vertex exactly once.

TRIE

A Trie, also called digital tree or prefix tree, is a kind of search tree—an ordered tree data structure used to store a dynamic set or associative array where the keys are usually strings.

Drawbacks Of Greedy Algorithm

1. Greedy algorithms don't work for some problems.

2. Despite their simplicity, correct greedy algorithms can be subtle.

3. Using greed is not an automatic.

What does it mean when we say that a computer is a programmable device?

Programmable means that data and instructions are logically the same and are stored in the same place. The consequence of this fact is that the program the computer executes is not wired into the hardware but entered from outside.

Pseudo Operation

A **pseudo**-**operation** describes the act of a software sending instructions or code to a hardware device that has not been compiled.

Recursive Algorithm

A recursive algorithm is an algorithm which calls itself with "smaller (or simpler)" input values, and which obtains the result for the current input by applying simple operations to the returned value for the smaller (or simpler) input.

Recursion

Recursion is a method of solving a problem where the solution depends on solutions to smaller instances of the same problem.

RAM Model

A ram Model has one processor and executed one instruction at a time each instruction is executed in a unit time.

Principles of RAM Model

Basic logical or arithmetic operations (+, *, =, if, call) are considered to be simple operations that take one time step.

• Loops and subroutines are complex operations composed of multiple time steps.

• All memory access takes exactly one time step.

Effectiveness Of An Algorithm

It must be possible to perform each step of the **algorithm** correctly and in a finite amount of time.

Advantages of Divide And Conquer

1. Always taking the best available choice is usually easy.

2. Repeatedly taking the next available best choice is usually linear work.

3. Much cheaper than exhaustive search

Master's Theorem

The master theorem for divide-and-conquer recurrences provides an asymptotic analysis for recurrence relations of types that occur in the analysis of many divide and conquer algorithms.

Tautology

This is a formula that is true in every possible interpretation.

Stringology

This is the study of algorithms and data structures used for processing text strings.

Cannon's Algorithm

Cannon's algorithm is a distributed algorithm for matrix multiplication for two-dimensional meshes first described in 1969 by Lynn Elliot Cannon. It is especially suitable for computers laid out in an N × N mesh.

Instruction Space

This is the amount of memory used to store compiled version of **instructions**.

Environmental Stack

This is the amount of memory used to store information of partially executed functions at the time of function call.

Data Space

This is the amount of memory used to store all the variables and constants.

Debugging An Algorithm

Algorithmic debugging is a debugging technique that compares the results of sub-computations with what the programmer intended.

Profiling An Algorithm

This is the use of **algorithms** or other mathematical techniques that allow the discovery of patterns or correlations in large quantities of data, aggregated in databases.

Linear Search

In this type of search, a sequential search is made over all items one by one. Every item is checked if a match is found then that particular item is returned, otherwise the search continues till the end of the data collection.

Phases For The Analysis Of An Algorithm

1. Priori Analysis:

This is a theoretical analysis of an algorithm. The efficiency of an algorithm is measured by assuming that all other factors are constant and have no effect on the implementation.

2. Posterior Analysis: This is an empirical analysis of the algorithm. The selected algorithm is implemented using programming language. This is then executed on the target computer machine.

Cryptography

Cryptography is the practice and study of techniques for securing communication and data in the presence of adversaries.

Classes of Cryptographic Algorithms

- Hash functions.
- Symmetric-key algorithms.
- Asymmetric-key algorithms.
- Symmetric-Key Algorithms for Encryption and Decryption.
- Message Authentication Codes (MACs)
- Digital Signature Algorithms.
- Discrete Logarithm based Key-Agreement Schemes.

Deterministic Algorithm

A **deterministic** algorithm is an algorithm which, given a particular input, will always produce the same output, with the underlying machine always passing through the same sequence of states.

Non Deterministic Algorithm

These are the algorithms in which the result of every algorithm is not uniquely defined and result could be random.

Legal Computation Of A Machine

A Legal Computation of the machine is a sequence of steps where in each step, the control unit enters a new state and the head writes new letter on the state which will move one step to the right or one step to the left justifying that the output can be negative or positive.

Cook's Theorem

This states that the Boolean satisfiability problem is NP-complete. That is, it is in NP, and any problem in NP can be reduced in polynomial time by a deterministic Turing machine to the Boolean satisfiability problem.

Fibonacci Number

The **Fibonacci sequence** is a set of **numbers** that starts with a one or a zero, followed by a one, and proceeds based on the rule that each number (called a **Fibonacci** number) is equal to the sum of the preceding two **numbers**.

Problem Development Steps

1. Problem Definition
2. Development of a model
3. Specification of an algorithm
4. Designing an algorithm
5. Checking correctness of an algorithm
6. Analysis Of An Algorithm
7. Implementation of algorithm
8. Program Testing
9. Documentation.

Boolean Satisfiability

A **formula** is said to be **satisfiable** if it can be made TRUE by assigning appropriate logical values (i.e. TRUE, FALSE) to its

variables. The **Boolean satisfiability** problem (SAT) is, given a **formula**, to check whether it is **satisfiable**.

Assumptions of RAM Model

- Each simple operation takes 1 time step.
- Loops and subroutines are not simple operations.
- Each memory access takes one time step, and there is no shortage of memory.

Assignments

Gauss Lemma For Polynomials

It follows from definitions that $I(PQ) \subset I(P)I(Q)$; thus if $I(PQ) = (1)$, then $I(P) = I(Q) = (1)$. It remains to prove the converse.

Suppose now that $I(P) = I(Q) = (1)$, and suppose $I(PQ) \neq (1)$. Then by Krull's Theorem, there is a (proper) maximal ideal $\mathfrak{m}$ that contains $I(PQ)$. Let $P = \sum_{k \geq 0} p_k x^k, \qquad Q = \sum_{k \geq 0} q_k x^k.$ Since $\mathfrak{m}$ is maximal, there exist least positive integers a, b such that $p_a, q_b \notin \mathfrak{m}$. Then the coefficient of x^{a+b} of PQ, $\sum_{k=0}^{a+b} p_k q_{a+b-k} \equiv p_a p_b \pmod{\mathfrak{m}},$ is not an element of $\mathfrak{m}$, since $\mathfrak{m}$ is a maximal, and therefore a prime, ideal. This is a contradiction. Therefore $I(PQ) = (1)$ if $I(P) = I(Q) = (1)$

1. Algorithm for cooking rice:

Method: English

Step 1: Fill a pot with two-thirds amount of water,

Step 2: Light a gas cooker and place the pot on it,

Step 3: Pour rice into a bowl and wash it three times,

Step 4: Pour washed rice into the pot on the cooker,

Step 5: Cover the pot and wait for 20 minutes,

Step 6: Open the pot and pour one and half tablespoon of salt in it,

Step 7: Place the lid to cover 70 percent of the pot,

Step 8: Wait twenty minutes,

Step 9: Turn off the cooker,

Step 10: Drain The rice.

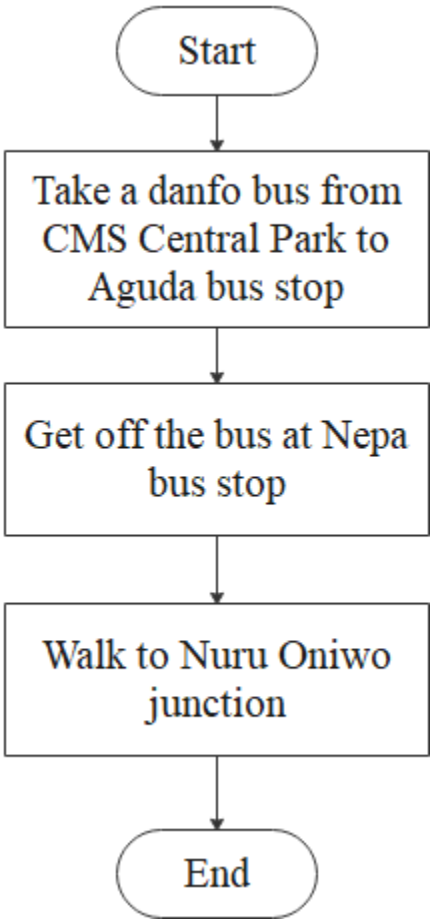
2. Algorithm to take you from CMS to Surulere:

Step 1: Take a danfo bus from CMS Central Park to Aguda bus stop,

Step 2: Get off the bus at Nepa bus stop

Step 3: Walk to Nuru Oniwo junction.

Flowchart to express the algorithm:


Start
Take a danfo bus from CMS Central Park to Aguda bus stop
Get off the bus at Nepa bus stop
Walk to Nuru Oniwo junction
End

Good Luck!