AL-402 Analysis & Design of Algorithms

S. No.	Date	Title	Page No.	Teacher's Sign / Remarks
1.	27 Feb 2023	Unit - 1 { Algorithm, Pseudocode, Flowchart	1 -	

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL

New Scheme Based On AICTE Flexible Curricula

CSE-Artificial Intelligence and Machine Learning/ Artificial Intelligence and Machine Learning
IV-Semester

AL402 Analysis &Design of Algorithms

Unit I : Definitions of algorithms and complexity, Time and Space Complexity; Time space tradeoff, various bounds on complexity, Asymptotic notation, Recurrences and Recurrences solving techniques, Introduction to divide and conquer technique, example: binary search, merge sort, quick sort, heap sort, strassen's matrix multiplicationetc, Code tuning techniques: Loop Optimization, Data Transfer Optimization, Logic Optimization, etc.

Unit II : Study of Greedy strategy, examples of greedy method like optimal merge patterns, Huffman coding, minimum spanning trees, knapsack problem, job sequencing with deadlines, single source shortest path algorithm etc. Correctness proof of Greedy algorithms.

Unit III : Concept of dynamic programming, problems based on this approach such as 0/1 knapsack, multistage graph, reliability design, Floyd-Warshall algorithm etc.

Unit IV : Backtracking concept and its examples like 8 queen's problem, Hamiltonian cycle, Graph colouring problem etc. Introduction to branch & bound method, examples of branch and bound method like travelling salesman problem etc. Meaning of lower bound theory and its use in solving algebraic problem, introduction to parallel algorithms.

Unit V : Advanced tree and graph algorithms, NP-hard and NP-complete problems, Approximations Algorithms, Data Stream Algorithms, Introduction to design and complexity of Parallel Algorithms.

References:

1. Coremen Thomas, Leiserson CE, Rivest RL, Introduction to Algorithms, Third edition, PHI.
2. Horowitz &Sahani, Analysis & Design of Algorithm, Fourth Edition Computer Science Press.
3. Dasgupta, algorithms, Fifth Edition, TMH
4. Ullmann; Analysis & Design of Algorithm, Addison-wesley publishing company,
5. Michael T Goodrich, RobartoTamassia, Algorithm Design, Wiely India
6. Rajesh K Shukla: Analysis and Design of Algorithms: A Beginner's Approach; Wiley

List of Experiments :

1. Write a program for Iterative and Recursive Binary Search.
2. Write a program for Merge Sort.
3. Write a program for Quick Sort.
4. Write a program for Strassen's Matrix Multiplication.
5. Write a program for optimal merge patterns.
6. Write a program for Huffman coding.
7. Write a program for minimum spanning trees using Kruskal's algorithm.
8. Write a program for minimum spanning trees using Prim's algorithm.
9. Write a program for single sources shortest path algorithm.
10. Write a program for Floye-Warshal algorithm.
11. Write a program for traveling salesman problem.
12. Write a program for Hamiltonian cycle problem.

AL-402
Analysis & Design of Algorithm

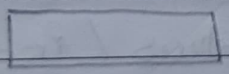
Unit - 1

Algorithm - * Collection of rules to solve a problem step by step
* Could be in natural language
* Never is programming language specific
* Never executed on machine.
* We have to dry run it ourselves.
* Representated through :- i> Flowcharts
ii> Pseudocode.

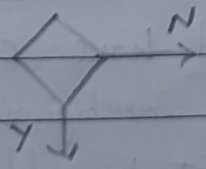
Flowcharts - * Pictorial representation of an algorithm using ~~through~~ shapes or symbols
* Problem is solved in stages.

Symbols used in Flowcharts :-


start / stop


step


Action / Process



Decision / condition

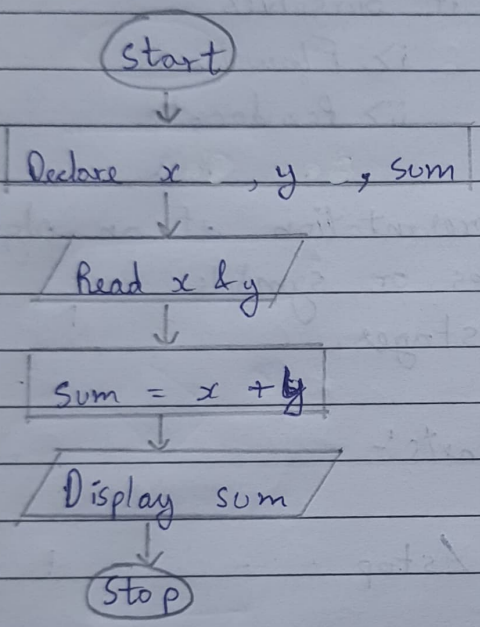

Flow line

Pseudo code - * Elaborates / explains each & every line unlike algorithm

* Pseudocode is informal, has no format whereas algorithm is formal way to explain solution for a problem.

Example :- Write an algorithm for addition of two numbers

Flowchart



Pseudo code

```
Begin // start the code
Declare x, y, sum // we are
// taking two variable
// to store diff values
// & a var to store the sum
sum = x + y // adding the
// two numbers
// & storing in sum
display sum // displaying the output
Stop // ending program.
```

Characteristics of an Algorithm / Features / Properties

1. Input → Atleast 1 input should be provided
2. Output → Atleast 1 output should be obtained
3. Definiteness → No ambiguity, should have clarity
   Eg: 'Add 10 or 20 to a' is ambiguous as the system doesn't know what to add 10 or 20.
   '10 / 0' is ambiguous as system doesn't know division by 0.
4. Finiteness → Program should get terminate somewhere It must have an end / halt.
5. Effectiveness → Accuracy, complexity / efficiency, should be able to dry run & perform tracing.

Algorithm

* A technique or collection of rules that describes how program is to be performed is known as algorithm

OR

It is a formal definition with some specific characteristic that describes a process which could be executed by the computer machine to perform a specific task.

* An algorithm helps to simplify and understand the problem.
It is an arrangement of steps to solve a problem.

* An algorithm is an efficient method that can be expressed within finite amount of time & space.

eg :-
i> start
ii> Read a, b
iii> c = a + b
iv> display c
v> stop

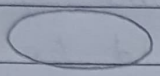
Flowcharts

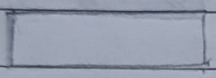
* Flowchart is a diagram that depicts the stages of a software or an application that takes to process data.

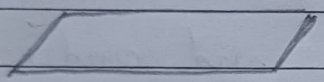
* Several geometric patterns can be used to show the different task for any software or application.

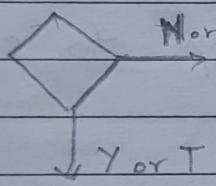
* It provides step-by-step direction to solve the problem.

* We use various symbols or shapes in order to depict different actions :-

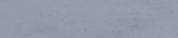
i>  → Start / Stop

ii>  → Processing function of program

iii>  → Input / output

iv> 

→ Decision making / branching

v>  → flowline

Pseudo code

* It is a non-standard manner to write a program.
* It only displays algorithm in English & mathematical notation.
* There is no specific programming language to create pseudocode. No grammar has to be followed. Furthermore, basic english language expression can be used to create pseudocode.
* It is something that lies between fully-Flegged simple syntactically correct code & well formed

grammatically correct sentences that describes the algorithm & the working of the program.

* It is occassionally used as a detailed phase in the development of a program.

* It allows designers to explain code in great detail & give programmers a thorough template or blueprint for developing code in a specific programming language.

* It is an informal way of writing a program. It is a method of developing an algorithm or it is basically a method of writing an algorithm.

eg → Begin // start the code

Accept First_num, second_num
// Input consists of accepting two numbers.

Compute sum is first_num + second_num
// Process of adding two numbers

Dislay sum of given two numbers // output for displaying sum

End // ending the pseudocode.

* Pseudo code uses Mnemonics.

Date 27/2/23
Page 6

Differentiate b/w pseudocode, flowchart & algorithm

	Algorithm	Flowchart	Pseudocode
1>	Instructions to address general problems	Pictorical representation	Instructions to address specific problems.
2>	Not a tool for creating documents.	Visual representation of an algorithm	Documents that can be represented by the tool algorithm
3>	Pure english & mathematical notation with no writing rules.	Represented by shapes & symbols.	In english but do not follow any set of rules

Properties / Characteristics / Features of Algorithm

(1) Input - Whatever raw that is provided to system to work upon it is known as input to system

"Each algorithm should be provided with atleast one or more inputs".

(2) Output - After processing the raw in the system the outcome of system is known as output.

"Atleast one output must result from an algorithm"

Algos give approximate results never accurate results.
Adv of algo → modifiability

classmate
Date 27/2/23
Page 7

(3) Definiteness - Each instruction of algorithm should be clear and unambiguous and should provide the finite result.

(4) Finiteness - Each algorithm must contain finite number of statements. In other words the algorithm must terminate somewhere.

(5) Effectiveness - Each algorithm must be simple to understand and can be run or executed manually with the help of pen & paper.

Problem development steps / Algorithm Analysis

Following steps are involved in solving computational problems:-

[Note : Computational problems are problems that calculate amount of resource required to solve that problem]

(1) Problem definition

(2) Development of model:

* Two common tools used for algorithm analysis are
  i) RAM model of computation
  ii) Asymptotic analysis of worst case complexity.

i) * RAM model of computation - Random Access Machine model is used for analysing algorithm without running them on a physical machine.

The RAM model has the following properties :-

a) A simple operation (+, -, *, /, if else ... etc) takes

by default RAM → cuz we don't know ... when writing algo

1 time step.

b> Loop & subroutines are comprised of simple operation.

c> Memory is unlimited & access takes 1 time step. (RAM model doesn't consider whether data is on cache or in disk)

d> Using RAM model you measure the running time of an algorithm by counting the number of steps an algorithm takes on a given input.

ii> Asymptotic Analysis

a> Best Case complexity - It is minimum number of steps to complete the code.

b> Worst case complexity - It is maximum no. of steps taken to complete the code.

c> Average case complexity - Avg no. of steps taken to complete.

(3) Specification of algorithm
→ either in step-by-step form or in pseudocode

(4) Designing an algorithm

(5) Check correctness of an algorithm

(6) Analysis of algorithm.

(7) Implementation of algorithm

(8) Program Testing.

(9) Documentation.

Date 27/2/23
Page 9

Q> What do you understand by Algorithm Design?

* It includes creating an efficient algorithm to solve a problem in an efficient way.
Some of them problem can be efficient w.r.t time consumption whereas other approaches may be memory efficient.
* However, both time consumption & memory usage cannot be optimized simultaneously.
If we required to run in lesser time so we have to invest more money in memory and if we require to run an algo with lesser memory then time consumption will be more

Q> Why analysis of algorithm is important.

i> To predict the behaviour of an algorithm without implementing it on specific computer.

ii> It is impossible to predict the exact behaviour of an algo. There are two many influencing factors.

iii> The analysis is just an approximation, it is not perfect.

iv> By analysing different algos, we can compare them to determine the best one for our purpose.

Types of algorithm techniques

① Recursive
② Divide & conquer
③ Greedy
④ Dynamic Programming
⑤ Backtracking
⑥ Branch & bound
⑦ Brute force