Mathematical Problem Solving

Teacher
Dr. Sasmita Kumari Nayak

Category

Course Name : Mathematical Problem Solving

Code(Credit) : CUTM1037(2-2-0)

Course Objectives

- To understand and analyze algorithms
- To understand efficiency of algorithms and alternative approaches
- To understand data structures and major algorithms and how they together play a role in efficiency
- To apply important algorithm design techniques using a programming language

Course Outcomes

COs	Course Outcomes	Mapping COs with POs (High-3, Medium-2, Low-1)
CO1	Ability to get knowledge on appropriate data type and data structure for a given problem	PO1 (3), PO12(1)
CO2	Analyze to compare algorithms with respect to time and space complexity	PO2(3)
CO3	Ability to select the best algorithm to solve a problem by considering various problem characteristics, such as the data size, the type of operations, etc	PO2(3)
CO4	Design and develop different algorithms for solving the problems	PO3 (3)

Course Syllabus

Module I: Introduction (4 Hrs)

Mathematics in Computer Science, Problem Solving and Algorithms, Data Structures and Algorithms, Algorithm Efficiency and Importance.

Module III: Algorithm Efficiency (6 Hrs)

Orders of Growth, Best-Case, Worst-Case and Average-Case Efficiencies, Analysis of Recursive and Non-Recursive Algorithms

Module IV: Brute Force and Exhaustive Search (10 Hrs)

Selection Sort and Bubble Sort, Sequential Search and Brute-Force String Matching, Exhaustive Search, Depth-First Search and Breadth-First Sea

Module V: Divide and Conquer (10 Hrs)

Merge Sort, Quick Sort, Binary Tree Traversal, Closest-Pair and Convex-Hull Problems

Module VI: Dynamic Programming and Greedy Technique (16 Hrs)

Dynamic programming- Floyd 's algorithm, Optimal Binary Search Trees, Knapsack Problem and Memory functions.

Greedy Technique- Prim'sAlgorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Trees.

Module VII: Limitations of Algorithm Power (4 Hrs)

Lower-Bound Arguments, Decision Trees, P, NP and NP- Complete Problems.

Text Books:

1. Fundamentals of Computer Algorithms, 2nd Edition, Ellis Horowitz, Sartaj Sahni and S. Rajasekharan
2. Introduction to the Design and Analysis of Algorithms, Anany Levitin

Reference Books:

1. Design and Analysis of Algorithms: S. Sridhar
2. Design and Analysis of Algorithms: P. H. Dave, H. B. Dave
3. Algorithm Design: Foundations, Analysis and Internet Examples: M. T. Goodrich and R. Tomassia, John Wiley and sons
4. Algorithms Illuminated (Part 2): Graph Algorithms and Data Structures:Tim Roughgarden

Session Plan

Session 1 & 2 (Module I)

Mathematics in Computer Science, Problem Solving and Algorithms:PPT

https://www.youtube.com/watch?v=BP8UzIZj-nU

https://www.youtube.com/watch?v=0IAPZzGSbME&list=PLch-RuUgf43mvNLxnsYjXffEG0fRdQ5KB

Session 3 & 4

Data Structures and Algorithms, Algorithm Efficiency and Importance: PPT

https://www.youtube.com/watch?v=0vz54KoZo9o

https://www.youtube.com/watch?v=tEI9XrxXhwY

Session 5 & 6 (Module II)

Sorting, Searching, String Processing:PPT

https://www.youtube.com/watch?v=KJWqhaejIn4

https://www.youtube.com/watch?v=SC5IH9GpYVo

Session 7 & 8

Graph and Numerical Problems: PPT
https://www.youtube.com/watch?v=1n5XPFcvxds&list=PLqM7alHXFySEaZgcg7uRYJFBnYMLti-nh
https://www.youtube.com/watch?v=0u78hx-66Xk&list=PLqM7alHXFySEaZgcg7uRYJFBnYMLti-nh&index=2

Session 9 & 10 (Module III)

Orders of Growth, Best-Case, Worst-Case and Average-Case Efficiencies: PPT
https://www.youtube.com/watch?v=D9SlLpPQRK8
https://www.youtube.com/watch?v=lj3E24nnPjI

Session 11 & 12

Analysis of Recursive algorithms: Substitution method, recursion tree method and master method: PPT
https://www.youtube.com/watch?v=JPAA1FbM7jk
https://www.youtube.com/watch?v=LC96q_6r3BQ
https://www.youtube.com/watch?v=OynWkEj0S-s

Session 13 & 14

Analysis of Non-Recursive Algorithms with example: PPT
https://www.youtube.com/watch?v=4HkIai9oiT0
https://www.youtube.com/watch?v=xFRea6xyPgM

Session 15 (Module IV)

Selection Sort: Problem-solving, Algorithm, Find Best, worst, and average case Time complexity: PPT
https://www.youtube.com/watch?v=8ps1XQigk7U
https://www.youtube.com/watch?v=peLS-S23TvE

Session 16

Practice 1:

- Sort the list E, X, A, M, P, L, E in alphabetical order by selection sort.

Session 17

Bubble Sort: Problem-solving, Algorithm, Find Best, worst, and average case Time complexity:PPT
https://www.youtube.com/watch?v=peLS-S23TvE
https://www.youtube.com/watch?v=04bA00_gFBU

Session 18

Practice 2:

- Sort the list E, X, A, M, P, L, E in alphabetical order by Bubble sort.

Session 19 & 20

Sequential Search and Brute-Force String Matching, Exhaustive Search:PPT
https://www.youtube.com/watch?v=Fit5lRYZAm4
https://www.youtube.com/watch?v=SX5vmCXdg5k
https://www.youtube.com/watch?v=zQyy7s4Lp6E

Session 21

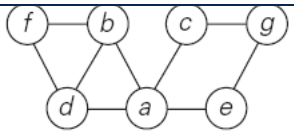
Practice 3:
You are given a string “s” and s pattern “p”, you need to check if the pattern is there in the string.
S = “prodevelopertutorial”
P = “rial”
We need to check if “rial” is present in “prodevelopertutorial” string. Use Brute Force Search Approach

Session 22 & 23

Depth-First Search and Breadth-First Search: problem solving, Find Time complexity:DFS-PPT, BFS-PPT
https://www.youtube.com/watch?v=pcKY4hjDrxk
https://www.youtube.com/watch?v=vf-cxgUXcMk

Session 24

Practice 4:



- Write down the adjacency matrix and adjacency lists specifying this graph. (Assume that the matrix rows and columns and vertices in the a follow in the alphabetical order of the vertex labels.)
- Starting at vertex *a* and resolving ties by the vertex alphabetical order, traverse the graph by depth-first search and construct the corresp first search tree. Give the order in which the vertices were reached for the first time (pushed onto the traversal stack) and the order in whi became dead ends (popped off the stack).

Session 25, 26 & 27 (Module V)

Merge Sort: Problem-solving, Algorithm, Find Best, worst, and average case Time complexity: LECTURE NOTES
https://www.youtube.com/watch?v=mB5HXBb_HY8
https://www.youtube.com/watch?v=g1AwUYauqgg
Practice 5:

- Sort the list E, X, A, M, P, L, E in alphabetical order by Merge sort.

Session 28, 29 & 30

Quick Sort: Problem-solving, Algorithm, Find Best, worst, and average case Time complexity: LECTURE NOTES
Practice 6:

- Sort the list E, X, A, M, P, L, E in alphabetical order by Quick sort.

Session 31, 32, 33 & 34

Binary Tree Traversal, Closest-Pair and Convex-Hull Problems: Closest pair-convex hull problem
VIDEO 1, VIDEO 2, VIDEO 3

Session 35, 36 & 37 (Module VI)

Dynamic programming- Floyd 's algorithm : Problem-solving, Algorithm, Find Best, worst, and average case Time complexity:PPT
VIDEO 1,VIDEO 2
Practice 7:

- Solve the all-pairs shortest-path problem for the digraph with the following weight matrix:

$$\begin{bmatrix} 0 & 2 & \infty & 1 & 8 \\ 6 & 0 & 3 & 2 & \infty \\ \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & 2 & 0 & 3 \end{bmatrix}$$

Session 38 & 39

Write pseudocode for a linear-time algorithm that generates the optimal binary search tree from the root table.

VIDEO 1, VIDEO 2

Session 40 & 41

knapsack problem: Problem solving, Algorithm, Time complexity

VIDEO 1, VIDEO 2, VIDEO 3

Practice 8:

a. Apply the bottom-up dynamic programming algorithm to the following instance of the knapsack problem:

item	weight	value
1	3	$25
2	2	$20
3	1	$15
4	4	$40
5	5	$50

capacity $W = 6$.

b. How many different optimal subsets does the instance of part (a) have?

c. In general, how can we use the table generated by the dynamic programming algorithm to tell whether there is more than one optimal subset knapsack problem's instance?

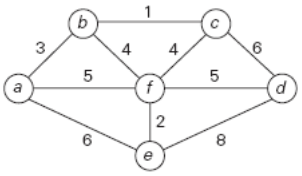
Session 42, 43, 44 & 45

Greedy Technique- Prim'sAlgorithm, Kruskal's Algorithm: Problem solving, Algorithms, Different cases of Time complexities: LECTURE NOTES

VIDEO 1, VIDEO 2, VIDEO 3, VIDEO 4

Practice 9 & 10:

a. Apply Prim's and Kruskal's algorithm to the following graph.:



Session 46 & 47

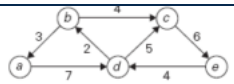
Dijkstra's Algorithm: Problem solving, Algorithms, Different cases of Time complexities: PPT

VIDEO 1, VIDEO 2

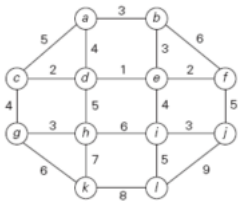
Practice 11:

Solve the following instances of the single-source shortest-paths problem with

vertex a as the source:



b.



Session 48 & 49

Huffman Trees: Problem solving, Algorithms, Different cases of Time complexities:LECTURE NOTES

VIDEO 1, VIDEO 2

Practice 12:

Consider the five-symbol alphabet {A, B, C, D, _} with the following occurrence frequencies in a text made up of these symbols:

symbol	A	B	C	D	_
frequency	0.35	0.1	0.2	0.2	0.15

Session 50, 51 & 52 (Module VII)

Lower-Bound Arguments, Decision Trees, P, NP and NP- Complete Problems:LECTURE NOTES 1, LECTURE NOTES 2, VIDEO

Assignment

https://drive.google.com/file/d/1bMs_xrrNb-zszGUeId_U3JscL1V_5_hv/view?usp=drive_link

Question Bank

https://drive.google.com/file/d/1qfHGpfa9EsQuvTHzT9PV1RwSkkLmayln/view?usp=drive_link

Centurion UNIVERSITY