

CS & IT ENGINEERING

Algorithms

Analysis Of Algorithms

Lecture No. - 01

By- Dr. Khaleel Khan
Sir

Topics to be Covered



Topic

Introduction to the Course



Hello, I'm Dr. Khaleel Ur Rahman Khan.

1. Ph.D. in Computer Science.
2. Professor in Computer Science.
3. Has more than 28 Years of Experience in Teaching at Engineering Colleges.
4. Published more than 50 journal articles in the areas of Wireless Networks.
5. Seven candidates have been awarded Ph.D. under his Supervision.
6. Has more than 22 years of Educating and Mentoring the GATE Aspirants.



By- Dr. Khaleel



My PW Official Telegram Channel

<https://t.me/KhaleelSirPw>



Topic : Lecture Schedule

Design & Analysis of
Algorithms

① ②



1. Analysis of Algorithms

- 1.1 Algorithm Concept and Lifecycle
- 1.2 Analysis of Algorithms
- * 1.3 Methodology & Types of Analysis
- * 1.4 (Asymptotic Notations) ASN 2/3 Q's
- * 1.5 Framework for Analysing Recursive Algorithms
- 1.6 Apriori analysis of Non-Recursive Algorithms
- * 1.7 Analysing Loops (loop complexities)
- 1.8 Space Complexity
- 1.9 Mathematical Background



Topic : Lecture Schedule

Design Strategies

2. Divide & Conquer

- 2.1 General Method
- 2.2 Max-Min Problem
- * 2.3 Merge Sort
- 2.4 Binary Search
- * 2.5 Quick Sort
- 2.6 Matrix Multiplication
- (*) 2.7 Long Integer Multiplication (LIM)
- (*) 2.8 * Master Method for D and C Recurrences
- 2.9 * Recursion Tree

Case Studies
Applications



Topic : Lecture Schedule



3. Greedy Method

- 3.1 General Method
- 3.2 Knapsack Problem
- 3.3 Job Sequencing with Deadlines
- 3.4 Optimal Merge Patterns
 - * 3.4.1 Huffman Coding
- 3.5 Minimum Cost Spanning Trees
 - * 3.5.1 Prims Method
 - 3.5.2 Kruskal's Method
- * 3.6 Dijkstras Shortest Paths Problem



Topic : Lecture Schedule

4. Dynamic Programming (DP)



4.1 The Method

* 4.2 Difference between DP, Greedy Method and DandC

4.3 Multistage Graphs

4.4 Travelling Salesperson Problem

4.5 Binary Knapsack Problem

4.6 All Pairs Shortest Paths

4.7 Bellman-Ford Single Source Shortest Paths

4.8 Longest Common Subsequence (LCS)

4.9 Matrix Chain Multiplication (MCP)

4.10 Sum of Subsets (SOS)

4.11 Reliable System Design

4.12 Optimal Cost Binary Search Tree

Applications



Topic : Lecture Schedule

5. Graph Algorithms

5.1 Representation of Graphs

5.2 Graph Traversals

 **DFS**

5.2.1 Undirected Connected Graphs

5.2.2 Undirected Disjoint Graphs: DFT

5.2.3 Directed Graphs & Types of Edges

5.2.4 DAG

BFS

5.2.5 FIFO BFS

5.2.6 LIFO BFS

5.2.7 LC BFS

5.3 Parenthesization Theorem



Topic : Lecture Schedule



6. Heap Algorithms

6.1 Operations : Create, Insert, Delete, Modify

6.2 Applications : Heapsort



Topic : Lecture Schedule



7. Sets

7.1 Representations

7.2 Operations





Topic : Lecture Schedule



8. Sorting Algorithms

8.1 Basic terminology

8.2 Methods

8.2.1 Bubble Sort

8.2.2 Selection Sort

8.2.3 Insertion Sort

8.2.4 Radix Sort



Topic : Lecture Schedule



9. Backtracking & Branch and Bound

Text – Books :

1. Introduction to Algorithms – Cormen (LCS Book)
2. Fundamentals of Algorithms – Horowitz and Sahni

Weightage : 8-10 m
(12-14 m)

Prerequisites:

1. Programming

[Constructs: if-then-else
loops

2. Data Structures:

[Stack + Queue + L.L

+ Trees +
Graphs]

3. Mathematics:

→ Functions

→ Series: (AP + GP + H.P)

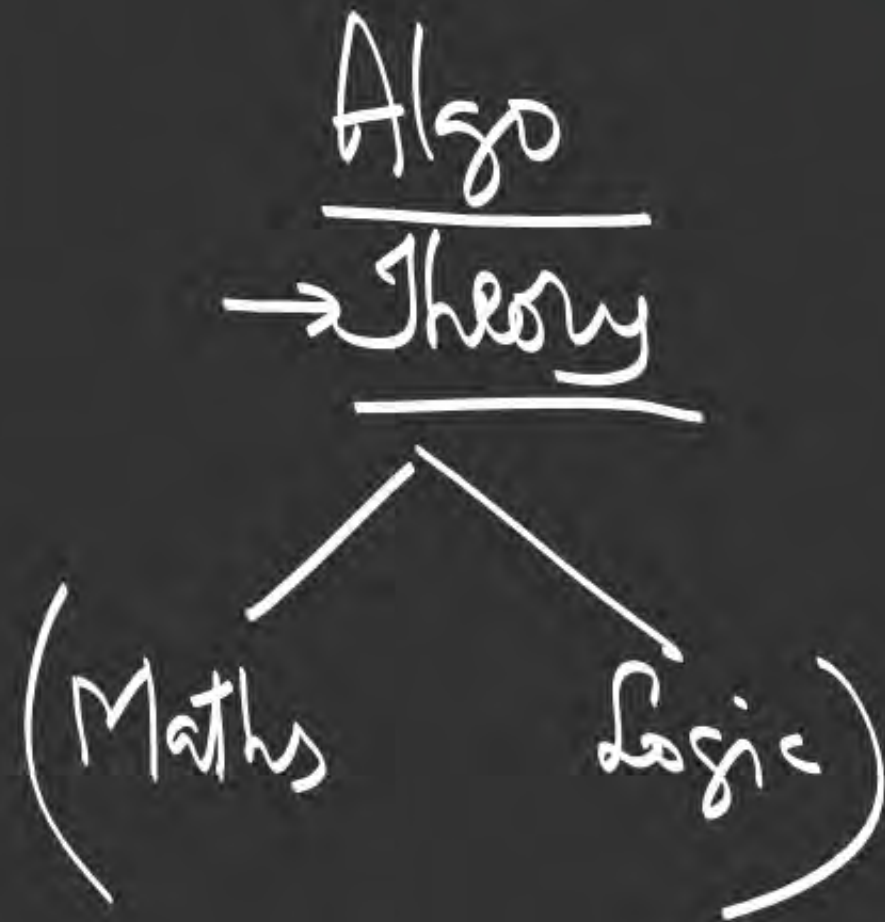
→ Matrices

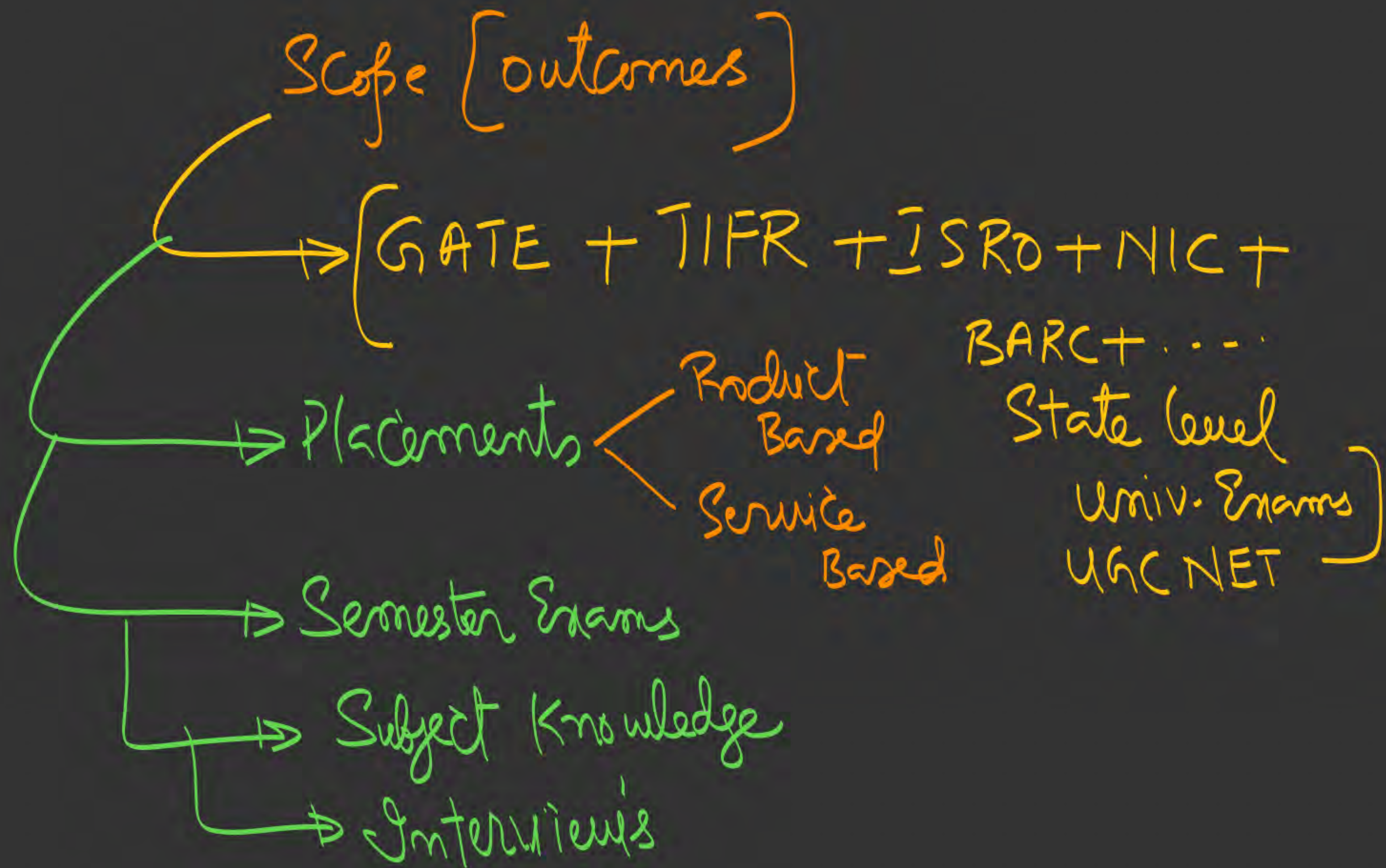
→ Logarithms

→ Exponents

→ Calculus

→ Probability





Algorithm



Other Courses:

- 1) A.I
- 2) Machine Learning
- 3) ~~Deep~~ " "
- 4) N.L.P
- + 5) Research

Algorithm Concept: Consists of Finite Set of Steps
(Solution) Statements
to solve a given Problem;

Persian: Md. Musa AlKhwarizmi

(Algorithm)

Consists of one/more
Fundamental operations

Ex: $(x \leftarrow y + z;)$ \leftarrow Addition (+)
Assignment (=)

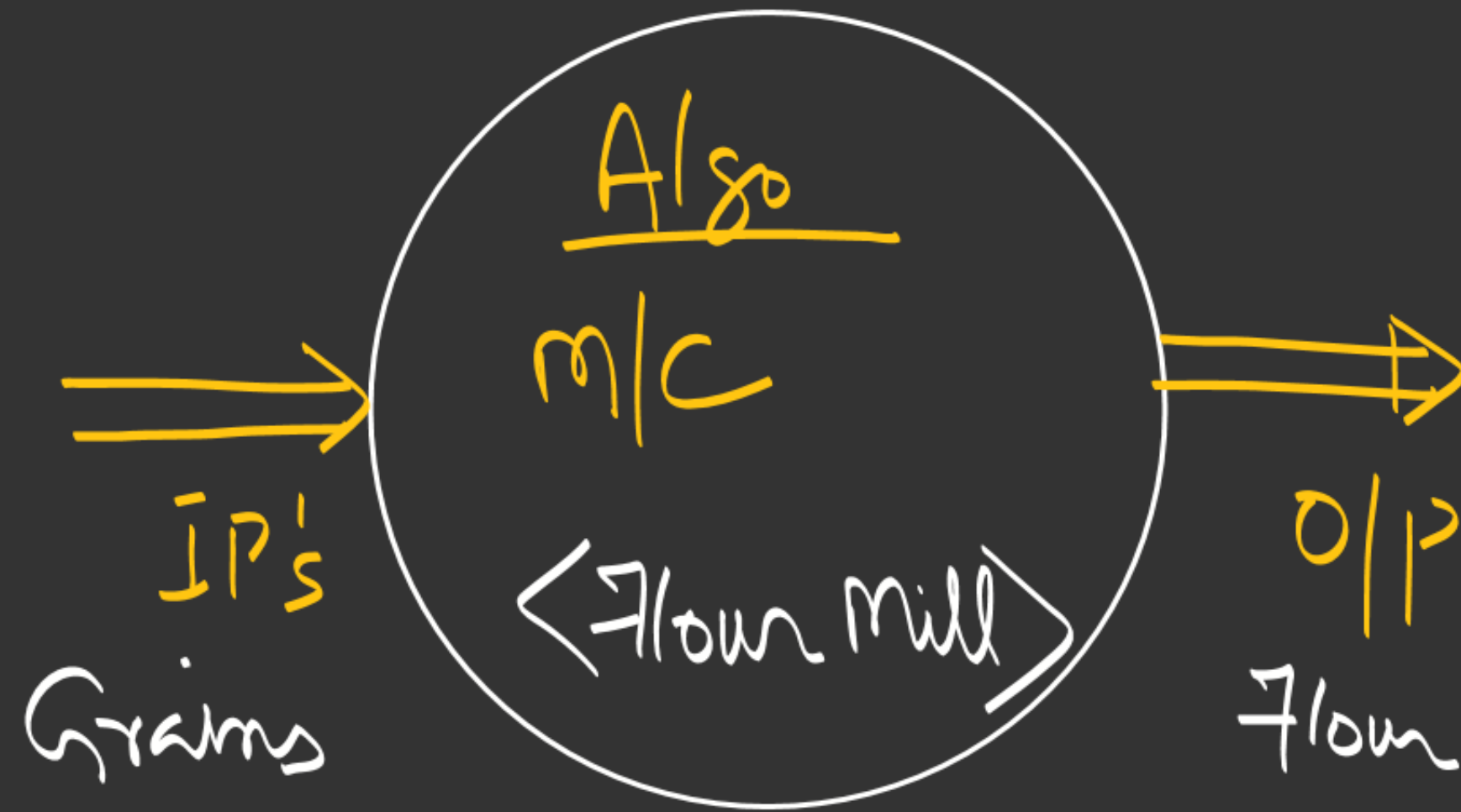
Definiteness (clear)
effective (finite time)

$(a + b *)$

→ Every Algorithm
may take zero/more
Inputs;

→ Every Algo. Must
Produce at least
one o/p

Algorithm ~ Abstract Machine/Processing Logic



Every Algo.
Should
Terminate
in, finite
amount
Time;

Algorithm Lifecycle Steps

1. Problem definition
2. Requirements (conditions) [SRS]
3. Logic / design
4. Develop Algorithm [Express]
5. Validation [Prove correctness]
6. Analysis
7. Implementation [Program development]
8. Testing & Debugging

DAA
→
(Design &
Analysis
of Algorithm)

THANK - YOU