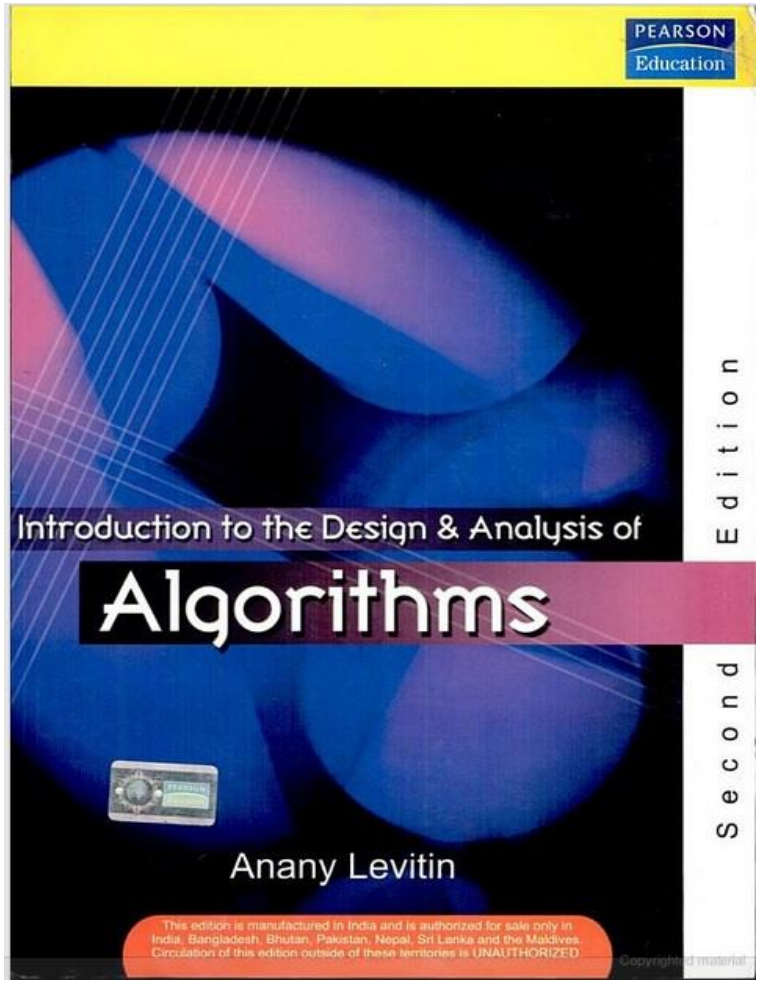
LOVELY PROFESSIONAL UNIVERSITY

CSE408
Design and analysis of algorithms

Lecture #0

Course details

- LTP – 3 0 0 [Three lectures/week]
- **(Self Learning Mode)**
- **Text Book**
  - INTRODUCTION TO THE DESIGN AND ANALYSIS OF ALGORITHMS
    - ANANY LEVITIN, PEARSON EDUCATION



nent Model

1. >=90% -- 5 marks

Each CA would be of 30 marks. Best 2 would be taken at the end

MTE would be of 40 marks

ETE would be of 70 marks and it would be prorated to 50 at the end

- ETE
- **Total**

Body presence

Mind Presence

5

20

25

50

100

Detail of Academic Tasks

- AT1: Test1 Before MTE
- AT2: Test2 Before MTE
- AT3: Test3 After MTE

Why we need to study this course ?

Why are we learning Design and analysis of algorithms?

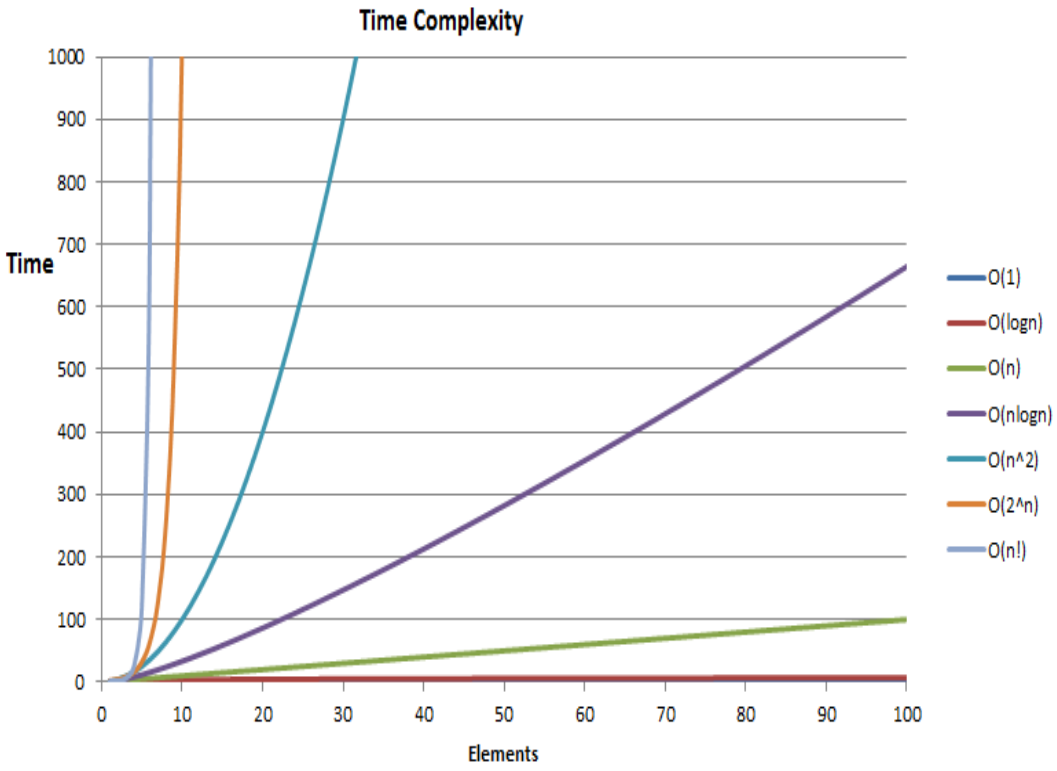
- Algorithms are used in almost every program or software system.
- Once we design an algorithm we need to know how well it performs on any input.
- In particular we would like to know whether there are better algorithms for the problem, An answer to this first demands a way to analyze an algorithm in a machine-independent way.
- Some Specific design techniques are essential ingredients of many software applications.

The course contents CSE408

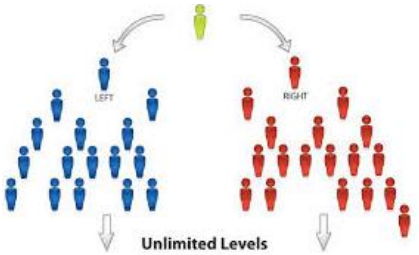
- UNIT I - Foundations of algorithm
- UNIT II - String matching algorithms and computational geometry
- UNIT III - Divide and conquer and ordered statists
- UNIT IV - Dynamic programming and greedy techniques
- UNIT V - Approximation algorithms
- UNIT VI - Number-theoretic algorithms and complexity classes

Foundations of algorithm

Analysis and growth



Basic DS



UNIT-1

- **Analysis of algorithm:**
- History and Motivation
- A Scientific Approach, Example: Quicksort
- Introductions to "big-oh" notation and asymptotic analysis.
- **Recurrence relations :**
- Computing Values
- Telescoping
- Types of Recurrences
- Mergesort
- Master Theorem .
- **Overview of generating functions:**
- Ordinary Generating Functions
- Counting with Generating Functions
- Catalan Numbers,
- Solving Recurrences
- Exponential Generating Functions
- **Asymptotics:**
- Standard Scale
- Manipulating Expansions
- Asymptotics of Finite Sums
- Asymptotics of Finite Sums
- **Trees:**
- Trees and Forests
- Binary Search Trees
- Path Length
- Other Types of Trees.

Link of Coursera:- https://www.coursera.org/learn/analysis-of-algorithms

UNIT II
String matching algorithms and computational geometry

UNIT-2

- Strings and Tries:
- Bitstrings with Restrictions
- Languages
- Tries
- Trie Parameters Key pattern matching concepts:
- Suffix Tree
- Suffix Array
- Knuth-Morris-Pratt algorithm .

Link of Coursera:- https://www.coursera.org/learn/algorithms-on-strings

Divide and conquer and ordered statists

UNIT-3

- What Are Divide and Conquer Algorithms?
- Max Subarray Problem Using Divide and Conquer
- Karatsuba's Multiplication Algorithm
- FFT Part 1: Introduction and Complex Numbers
- FFT Definition and Interpretation of Discrete Fourier Transforms
- FFT: Divide and Conquer Algorithm for FFT Application
- # 1 : Fast Polynomial Multiplication using FFT Application
- # 2: Data Analysis using FFT

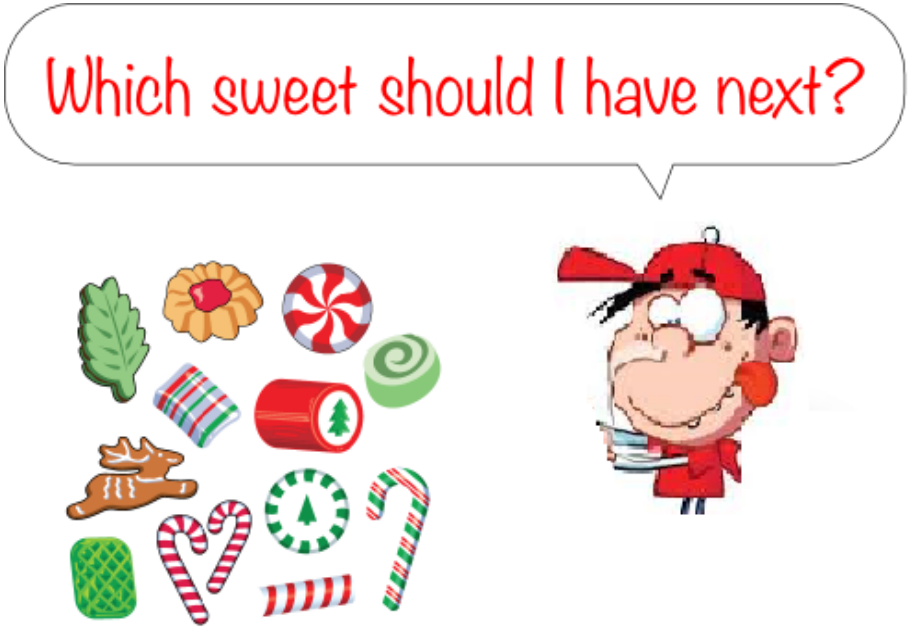
Link of Coursera

https://www.coursera.org/learn/dynamic-programming-greedy-algorithms#modules

LOVELY PROFESSIONAL UNIVERSITY

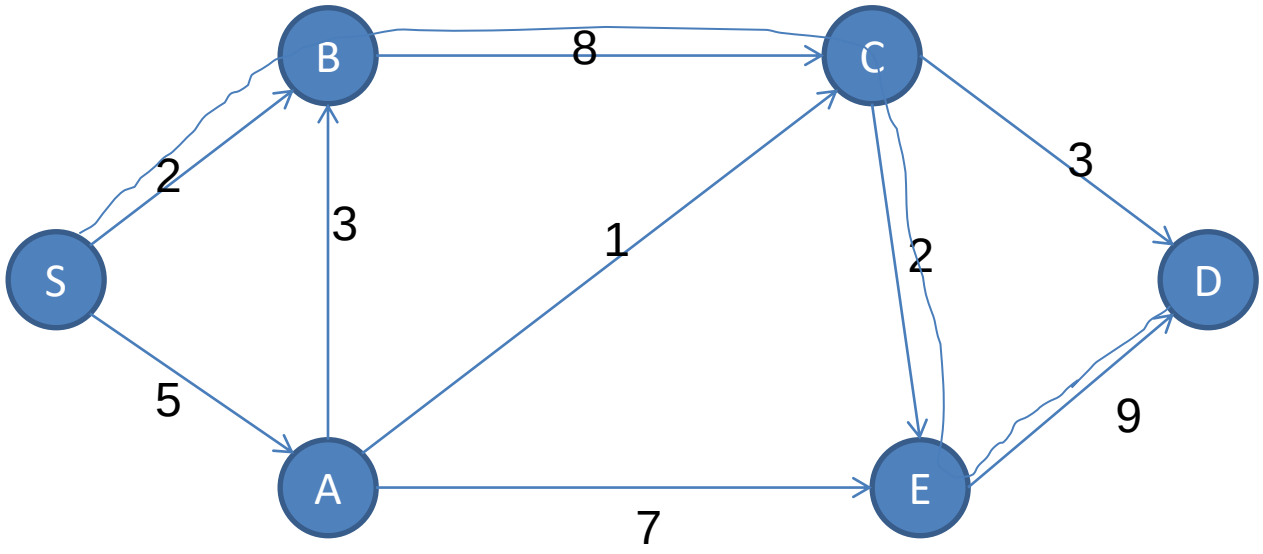
Dynamic programming and greedy technique

Dynamic programming(Knap sack)



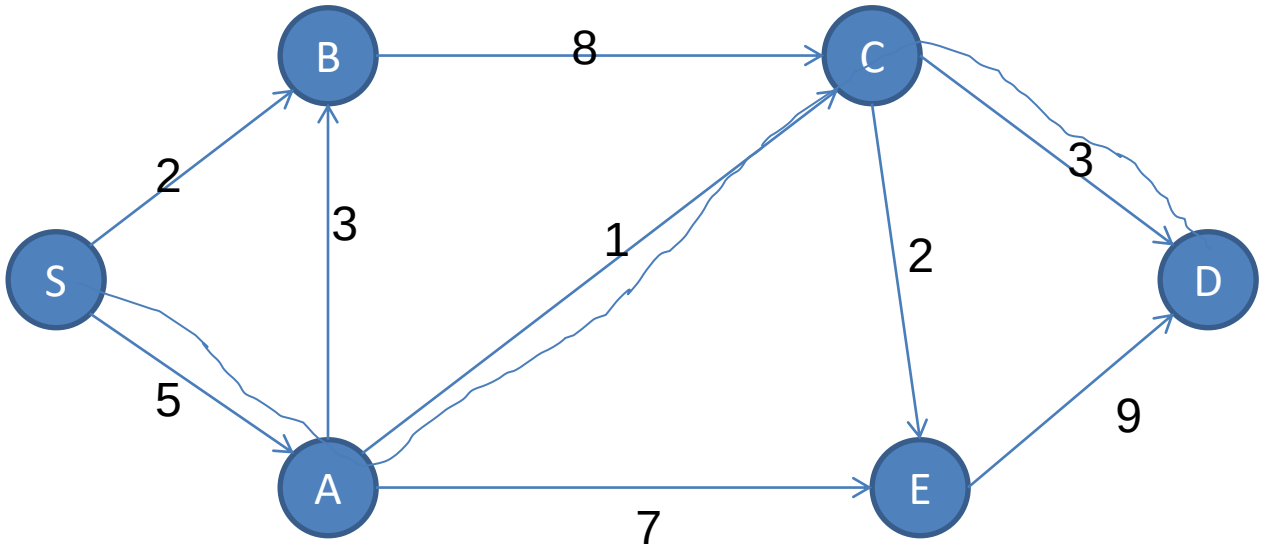
Greedy technique

Greedy technique



Shortest Distance=?

Dynamic programming



Shortest Distance=?

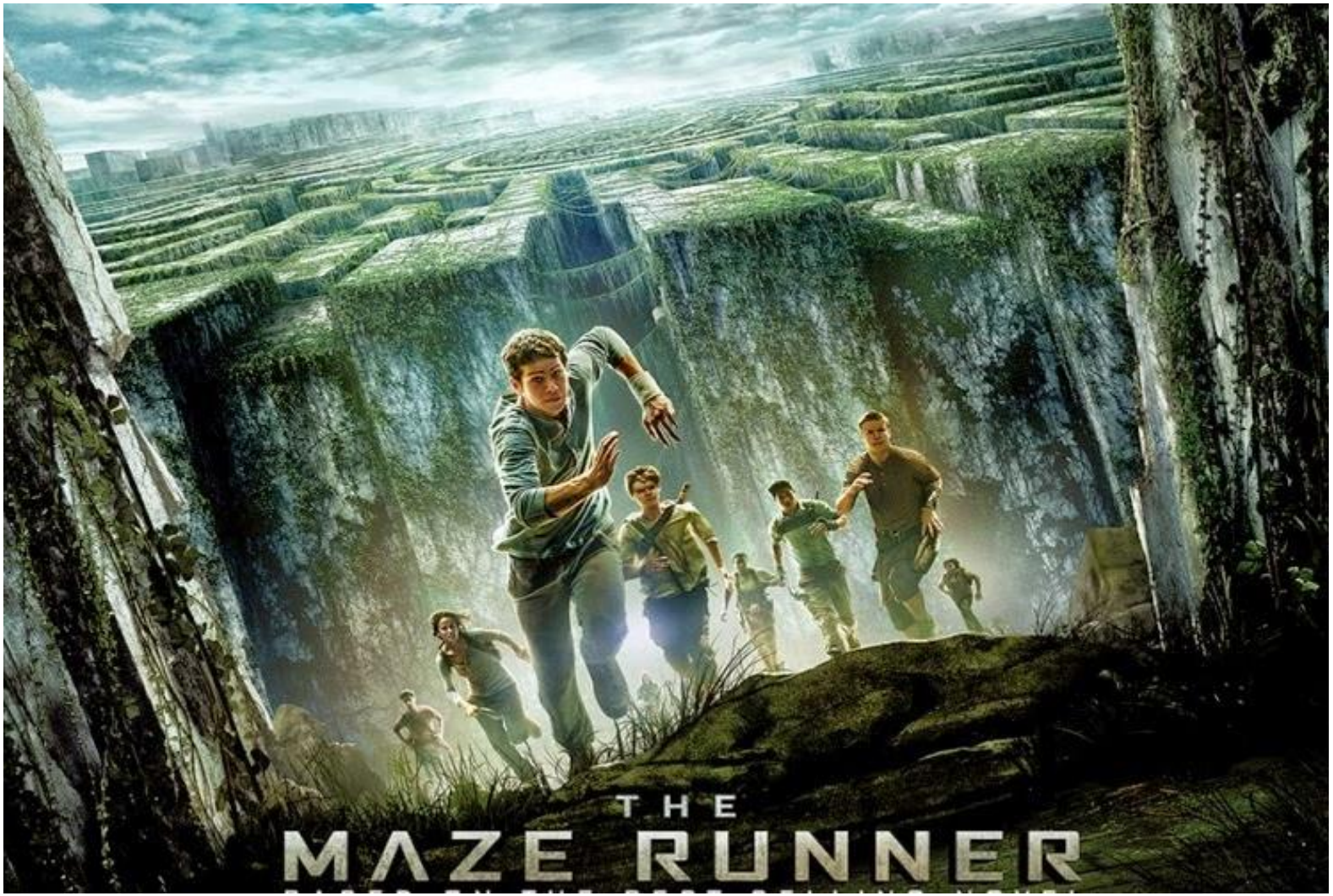
UNIT-4

- Introduction to Dynamic Programming + Rod Cutting Problem
- Coin Changing Problem
- Knapsack Problem
- When Optimal Substructure Fails
- Dynamic Programming: Longest Common Subsequence
- Memoization
- Coin Changing Problem.
- Introduction to Greedy Algorithms
- Greedy Interval Scheduling
- Prefix Codes
- Huffman Codes
- Huffman Codes: Proof of Optimality

Link of Coursera

https://www.coursera.org/learn/dynamic-programming-greedy-algorithms#modules

Backtracking and approximation algorithms



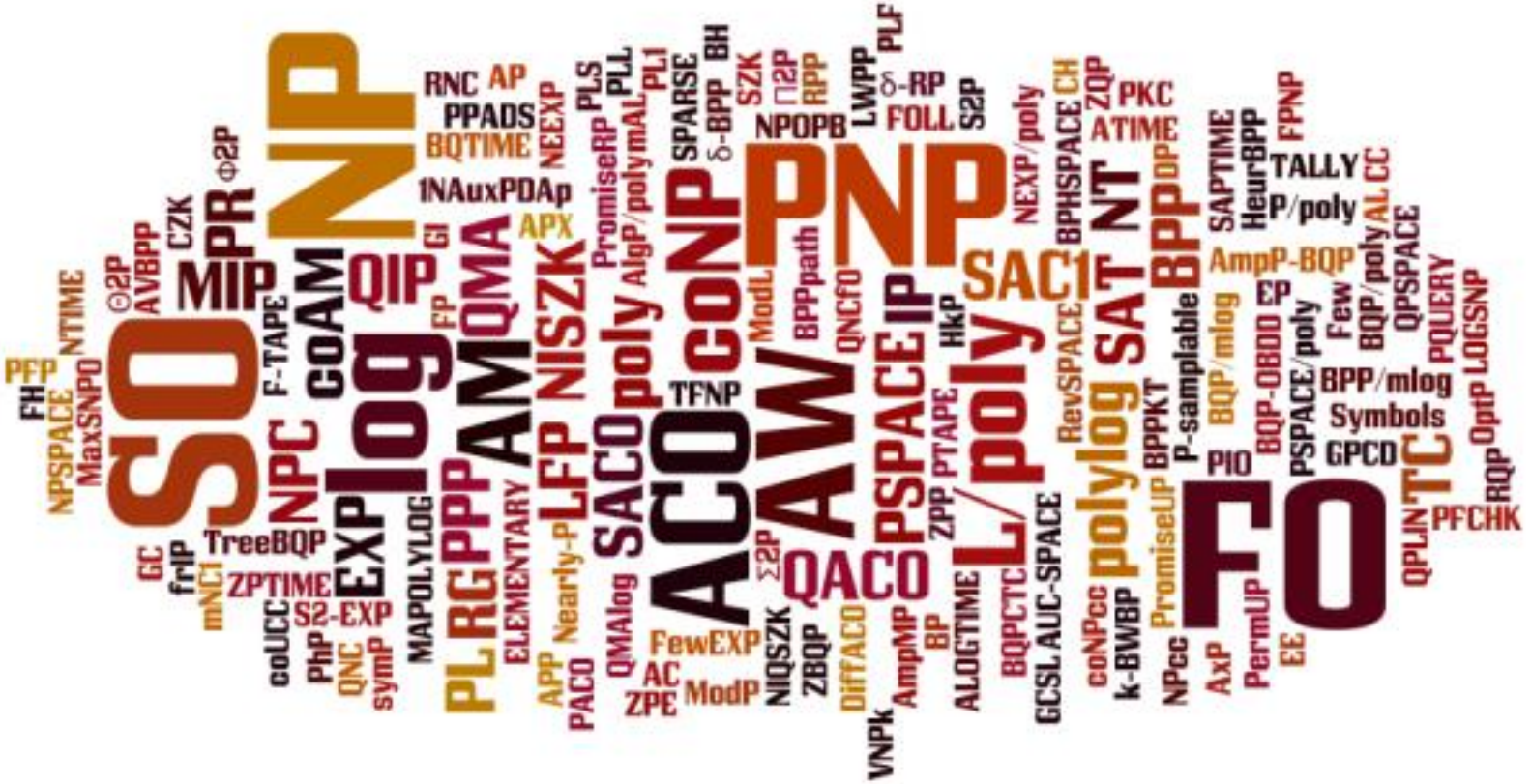
UNIT-5

- Introduction to Approximation Algorithms
- Introduction to Job shop Scheduling and Algorithm Design
- Analysis of Job shop Scheduling
- Approximation Algorithms for Vertex Cover and their Analysis,
- Approximation Algorithms for the Maximum Satisfiability Problem
- Travelling Salesman Problem
- Approximation Schemes: Introduction to TSP and its applications
- NP-Hardness of TSPs
- Hardness of Approximating General TSPs

Link of Coursera
https://www.coursera.org/learn/linear-programming-and-approximation-algorithms#modules

LOVELY PROFESSIONAL UNIVERSITY

Number-theoretic algorithms and complexity classes



UNIT-6

- Decision Problems and Languages
- Polynomial Time Problems
- NP Definition
- NP Completeness and Reductions
- NP Complete Problems: Examples
- Computation and Physics Qubits and Operations
- Bell's Inequality
- Grover's Search Algorithm

Link of coursera

https://www.coursera.org/learn/dynamic-programming-greedy-algorithms#modules

- **It is mandatory to complete the number of course for being eligible for End Term Examination along with the attendance criteria of the university. The links of the courses as shared in the IP should be completed on/before the last teaching day as per the academic calendar of the university.**

What will be the course outcome?

CO1: Understand the basic techniques of analyzing the algorithms using space and time complexity, asymptotic notations

CO2: Apply the various string matching algorithms

CO3: Analyze the divide and conquer algorithm design technique using various searching and sorting algorithms

CO4: Evaluate the various dynamic programming and greedy algorithm design technique to solve various problems

CO5: :Apply the backtracking method to solve some classic problems and understand branch and bound algorithm design technique

CO6: Define Number-Theoretic Algorithms and Complexity Classes and understand the basics concepts of complexity classes

What will be the Program outcome?

PO1::Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2::Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.

PO3::Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4::Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5::Create, select , and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6::Apply reasoning informed by the contextual knowledge to assess societal , health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7::Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8::Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9::Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings .

PO10::Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

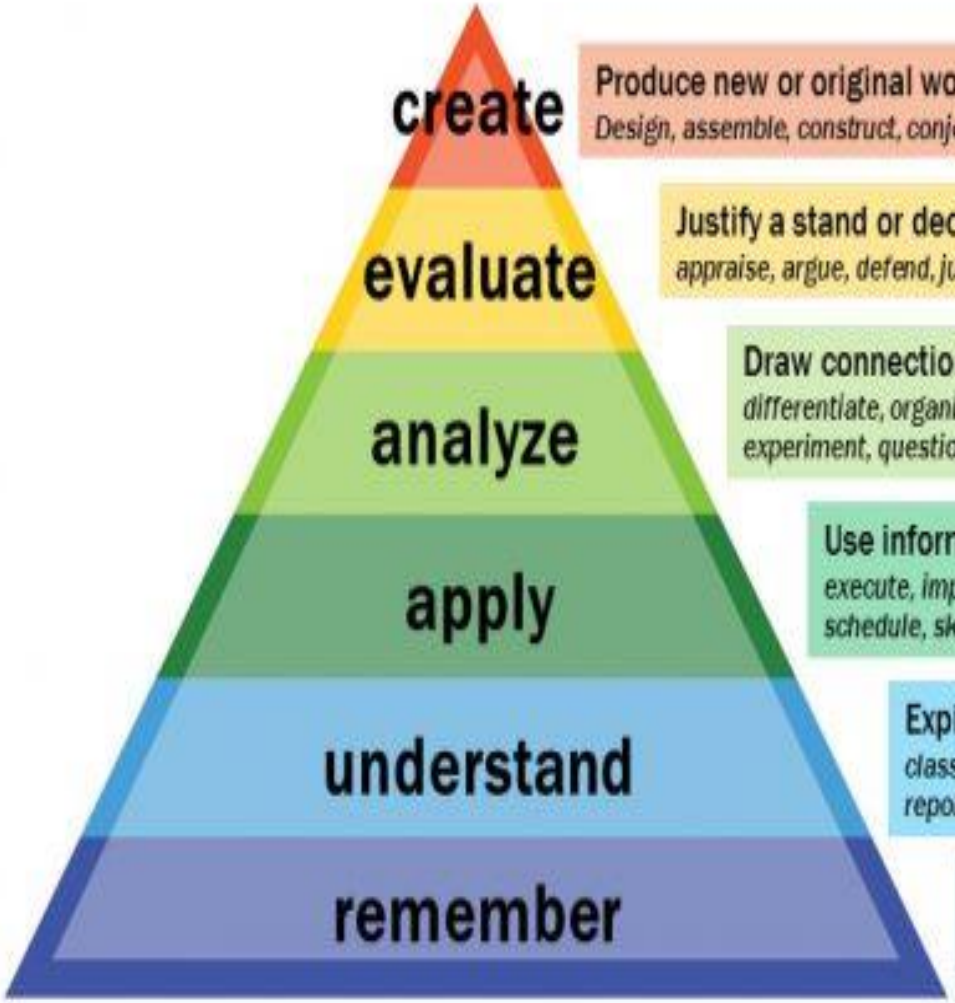
PO11::Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12::Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

MOOCs or Industry certification

Details of MOOC Provider	MOOC Name
Coursera	Design and Analysis of algorithms

Bloom's Taxonomy



Produce new or original work
Design, assemble, construct, conjecture, develop, formulate, author, investigate

Justify a stand or decision
appraise, argue, defend, judge, select, support, value, critique, weigh

Draw connections among ideas
differentiate, organize, relate, compare, contrast, distinguish, examine, experiment, question, test

Use information in new situations
execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch

Explain ideas or concepts
classify, describe, discuss, explain, identify, locate, recognize, report, select, translate

Recall facts and basic concepts
define, duplicate, list, memorize, repeat, state

OER(OPEN EDUCATIONAL RESOURCE)

OPEN EDUCATIONAL RESOURCE

Course Code: CSE408

Course Title: DESIGN AND ANALYSIS OF ALGORITHMS

L.T.P: 3.0.0 **Credit: 3**

Course Code	Course Title	Unit mapped	Broad topic/Sub Topic	OER Type	Title of OER	*%age unit mapped with OER (approx)	Source URL
CSE408:DESIGN AND ANALYSIS OF ALGORITHMS		Unit 1	Analysis of algorithm: History and Motivation, A Scientific Approach, Example: Quicksort. Introductions to "big-oh" notation and asymptotic analysis.Recurrence relations : Computing Values, Telescoping, Types of Recurrences, Mergesort , Master	online Coursera Portal	Modules and content will be made available through the online Coursera Portal, enabling individuals to attain a heightened clarity on algorithmic concept		https://www.coursera.org/learn/analysis-of-algorithms https://www.coursera.org/learn/data-structures

		Overview of generating functions: Ordinary Generating Functions, Counting with Generating Functions, Catalan Numbers, Solving Recurrences, Exponential Generating Functions. Asymptotics: Standard Scale, Manipulating Expansions, Asymptotics of Finite Sums, Asymptotics of Finite Sums Trees: Trees and Forests, Binary Search Trees, Path Length, Other Types of Trees.				
	Unit 2	Strings and Tries: Bitstrings with Restrictions , Languages, Tries , Trie Parameters Key pattern	online Coursera Portal	Modules and content will be made available through the online Coursera Portal, enabling		https://www.coursera.org/learn/analysis-of-algorithms

		matching concepts: Suffix Tree, Burrow wheeler transform and Suffix Array, Knuth-Morris-Pratt algorithm . .		individuals to attain a heightened clarity on algorithmic concept		
	Unit 3	What Are Divide and Conquer Algorithms? Max Subarray Problem Using Divide and Conquer Karatsuba's Multiplication Algorithm FFT Part 1: Introduction and Complex Numbers FFT Definition and Interpretation of Discrete Fourier Transforms FFT: Divide and Conquer Algorithm for FFT Application	online Coursera Portal	Modules and content will be made available through the online Coursera Portal, enabling individuals to attain a heightened clarity on algorithmic concept		: https://www.coursera.org/learn/dynamic-programming-greedy-algorithms#modules

		# 1 : Fast Polynomial Multiplication using FFT Application # 2: Data Analysis using FFT				
	Unit 4	Introduction to Dynamic Programming + Rod Cutting Problem Coin Changing Problem Knapsack Problem When Optimal Substructure Fails Dynamic Programming: Longest Common Subsequence Memoization Coin Changing Problem. Introduction to Greedy Algorithms Greedy Interval Scheduling Prefix Codes	online Coursera Portal	Modules and content will be made available through the online Coursera Portal, enabling individuals to attain a heightened clarity on algorithmic concept		https://www.coursera.org/learn/dynamic-programming-greedy-algorithms#modules

		Huffman Codes Huffman Codes: Proof of Optimality				
	Unit 5	Introduction to Approximation Algorithms,: Introduction to Job shop Scheduling and Algorithm Design, Analysis of Job shop Scheduling, Approximation Algorithms for Vertex Cover and their Analysis, Approximation Algorithms for the Maximum Satisfiability Problem, Travelling Salesman Problem and Approximation Scemes: Introduction to TSP and its applications, NP-Hardness of TSPs, Hardness of Approximating General TSPs, Held and Karp's Dynamic Programming Algorithm, Integer Linear Programming Formulation, Subtours and	online Coursera Portal	Modules and content will be made available through the online Coursera Portal, enabling individuals to attain a heightened clarity on algorithmic concept		Approximation Algorithms and Linear Programming | Coursera

		Subtour Elimination Formulation, Metric TSP and Shortcutting, Eulerian Walks for approximating TSPs, Christofides Algorithm and its Analysis, Heuristics for TSPs, Full Polynomial Time Approximation Scheme and Knapsack.				
	Unit 6	• Decision Problems and Languages • Polynomial Time Problems • NP Definition • NP Completeness and Reductions • NP Complete Problems: Examples • Computation and Physics Qubits and Operations • Bell's Inequality • Grover's Search Algorithm	online Courisera Portal	Modules and content will be made available through the online Coursera Portal, enabling individuals to attain a heightened clarity on algorithmic concept		https://www.coursera.org/learn/dynamic-programming-greedy-algorithms#modules

Let’s Start: Foundations of algorithm