Lesson Plan No. 1	Course Name: Design and Analysis of Algorithms	Course No.: COM-301

Topics	**Design and Analysis of Algorithms: Introduction to the Course**
Objectives	At the end of the lesson the student shall be able to: a. Understand the importance and application of algorithms in computer science. b. Recognize the basic concepts and terminologies related to algorithms. c. Identify various career opportunities that involve algorithm design and analysis. d. Describe the structure of the course and the expected outcomes.
Teaching Aids (if any)	a. Presentation Slides b. Whiteboard and Markers c. YouTube Video
Teaching Development	1. **Introduction** (5 minutes) **Ask questions:** ✓ Can you name an algorithm that you use in your daily life? ✓ Why do you think algorithms are important in technology? 2. **Development** (30 minutes) a. Real-Life Applications of Algorithms • Search engines (e.g., Google Search) • Social media feeds (e.g., Facebook, Instagram) • Navigation apps (e.g., Google Maps) b. Why Study Algorithms? • Role of Algorithms in Computer Science ✓ Backbone of programming and software development. ✓ Enables efficient data processing and problem-solving. • Benefits ✓ Improves performance and efficiency ✓ Helps in optimizing resources ✓ Essential for innovation and technological advancement c. Career & Job Perspective: Career Roles • Algorithm Engineer • Data Scientist • Software Developer d. Industry Demand

	• High demand for skills in algorithm design and analysis e. Certifications and Courses Coursera Link: **https://www.coursera.org/programs/test-program-93zov/learn/analysis-of-algorithms?source=search** f. Course Overview: Key Units and Topics ✓ Time and Space Complexity ✓ Sorting Algorithms ✓ Greedy Approach & Dynamic Programming ✓ Graph Algorithms & Backtracking g. Explain the Learning Outcomes (CO1 to CO5) of the Course. 3. **Short Animation Video** (5 minutes) **Video Link:** **https://www.youtube.com/watch?v=6hfOvs8pY1k&t=14s** 4. **Interactive Session** (5 minutes) • Future Trends in Algorithm Design : Emerging Fields ✓ Machine Learning ✓ Artificial Intelligence ✓ Quantum Computing • Impact : Innovations driven by new algorithms
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: **a. Introduction to Algorithms** by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition. **b. Algorithms** by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition. 3. **Next Lecture Topic : "Mathematical Preliminaries "**
Evaluation	Reflective Questions (What, Why, Who?). Allow students to answer and discuss. ✓ Can you explain why algorithms are fundamental to computer science? ✓ What are the benefits of using algorithms in technology? ✓ Describe a career role that involves working with algorithms. Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1.1	Course Name: Design and Analysis of Algorithms	Course No.: COM-301

Topics	**Design and Analysis of Algorithms: Mathematical Preliminaries**
Objectives	At the end of the lesson the student shall be able to: a. Understand Objectives of Learning Algorithms b. Explain significance of Preliminary Terms c. Describe Mathematical Model of a Computer d. Understanding the concept of Demonstrating Incorrectness. e. Reasoning about Correctness of an algorithm.
Teaching Aids (if any)	a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	1. **Introduction** (5 minutes) - Ask questions: What do you understand by an algorithm? - Define the term problem instance. Does the algorithm depend on any programming language? - Focus on the issues of algorithm correctness. - Have a discussion on the issue of finding the best possible answer or achieving maximum efficiency. - Introducing how Correct algorithms usually come with a proof of correctness. 2. **Development** (30 minutes) a. Understand Objectives of Learning Algorithms - Explain the tools to distinguish correct algorithms from incorrect ones. - Elaborate why Correct algorithms require careful exposition, and efforts to show both correctness and not incorrectness. b. Describe the Mathematical Model of a Computer - Explain the significance of the RAM Model and how It works. - Working Principles of EM Model. - PRAM Model working criteria along with advantages and application areas. c. Expressing Algorithms

	- Reasoning about why an algorithm is impossible without a careful description of the sequence of steps to be performed. - Listing the most common forms of algorithmic notation What is an algorithm? https://www.youtube.com/watch?v=cuhLSGGV-1k d. Summarize the Objectives of Learning Algorithms and Working on Various Mathematical Models of a Computer. 3. **Exercise** (10 minutes) – - Have a discussion to summarize the lecture - Ask Questions Related to Topics
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: **a. Introduction to Algorithms** by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition. **b. Algorithms** by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition. **Homework**: 3. **Activity**: Understanding the working of Mathematical Model of a Computer. **Activity References**: **a.** https://opendatastructures.org/versions/edition-0.1e/ods-cpp/1_3_Model_Computation.html **b.** https://medium.com/@_SD10_/the-ram-model-of-computation-and-big-o-notation-a1b3cc50ec2c Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	**1.** Reflective Questions (What, Why, Who?). Allow students to answer and discuss. **2.** Nearpod / Mentimeter Quiz on Objectives of Learning Algorithms and Working of various Mathematical Models of a Computer. **3.** MCQ / Sessional Test / Assignments Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1.2	Course Name: Design and Analysis of Algorithms	Course No.: COM-301

Topics	**Time complexity**
Objectives	At the end of the lesson, the student shall be able to: a. Analyze the time complexity of common sorting algorithms. b. Evaluate the efficiency of an algorithm in terms of its time complexity. c. Implement algorithms in a programming language and measure their time complexity experimentally. d. Understanding the time-space trade-off involving a problem. e. Solve examples on finding Time Complexity.
Teaching Aids (if any)	a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	1. **Introduction** (5 minutes) - Ask questions: What is the difference between an algorithm and a program? - Define the term pseudocode. - Focus on the issues of Running Time of an Algorithm. - Discuss the memory requirement of an algorithm. - Introducing Time- Space trade-off in an algorithm. 2. **Development** (30 minutes) a. Understand Objectives of Time Complexity of an algorithm - How Time complexity is profoundly related to the input size. - Explain the thought process behind innovation and technology used to make life easier for people by providing ways to solve problems they may encounter. - Elaborate why it is always better to select the most efficient algorithm when a simple problem can be solved with different methods. b. Describing what it Takes to Create a Good Algorithm?

	c. Explaining factors that play a significant role in the long-term usage of an algorithm. d. Time Complexity in Analysis of Algorithm https://www.youtube.com/watch?v=88Rq4U2mioQ e. Summarize the Objectives of Time complexity. 3. **Exercise** (10 minutes) – - Have a discussion to summarize the lecture - Ask Questions Related to Topics
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: **a. Introduction to Algorithms** by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition. **b. Algorithms** by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition. **Home work**: 3. **Activity** : Understanding the Significance of Time Complexity of an algorithm. **Activity References**: **a.** https://towardsdatascience.com/space-and-time-complexity-in-computer-algorithms-a7fffe9e4683 **b.** https://www.hackerearth.com/practice/basic-programming/complexity-analysis/time-and-space-complexity/tutorial/ Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	**1.** Reflective Questions (What, Why, Who?). Allow students to answer and discuss. **2.** Google form Quiz on the topic Time Complexity of an algorithm. **3.** MCQ / Sessional Test / Assignments Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1.3	Course Name: Design and Analysis of Algorithms	Course No.: COM-301

Topics	**Space complexity**
Objectives	At the end of the lesson the student shall be able to: a. Define space complexity and understand its importance in the context of algorithm design and analysis. b. evaluate how different data structures (e.g., arrays, linked lists, trees, graphs) impact the space complexity of algorithms. c. Analyze the space complexity of real-world algorithms and applications, such as those used in databases, networking, and machine learning. d. Identify the space complexity of basic algorithms, including sorting algorithms (e.g., bubble sort, merge sort) and searching algorithms (e.g., binary search). e. Solve examples of finding Space Complexity.
Teaching Aids (if any)	a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	1. **Introduction** (5 minutes) - Ask questions: What is the difference between an algorithm and a program? Define the term pseudocode. - Focus on the issues of Running Time of an Algorithm. - Have a discussion on the memory requirement of an algorithm. - Introducing Time- Space trade-off in an algorithm. 2. **Development** (30 minutes) a. Describe Space Complexity of an algorithm - Explain the significance of Auxiliary space and Input space. - Illustrating why the factor of time is usually more important than that of space. b. Describing what it Takes to Create a Good Algorithm? c. Explaining factors that play a significant role in the long-term usage of an algorithm. d. Space Complexity of an Algorithm

	https://www.youtube.com/watch?v=yOb0BL-84h8 3. **Exercise** (10 minutes) – - Have a discussion to summarize the lecture - Ask Questions Related to Topics
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: **a.** **Introduction to Algorithms** by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition. **b.** **Algorithms** by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition. **Home work**: 3. **Activity** : Understanding the Significance of Space Complexity of an algorithm. **Activity References**: **a.** https://towardsdatascience.com/space-and-time-complexity-in-computer-algorithms-a7fffe9e4683 **b.** https://www.hackerearth.com/practice/basic-programming/complexity-analysis/time-and-space-complexity/tutorial/ Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	**1.** Reflective Questions (What, Why, Who?). Allow students to answer and discuss. **2.** Google form Quiz on the topic Space Complexity of an algorithm. **3.** MCQ / Sessional Test / Assignments Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1.4	Course Name: Design and Analysis of Algorithms	Course No.: COM-301

Topics	**Worst-case analysis**
Objectives	At the end of the lesson, the student shall be able to: a. Understand how the worst case of an algorithm depends on the size of the user input value. b. Calculate the upper limit of the execution time of an algorithm. c. Understanding the concept of Worst-case analysis with the help of an example. d. Describe the concept of best-case analysis with the help of an example. e. Elaborate the concept of worst-case analysis with the help of an example.
Teaching Aids (if any)	a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	1. **Introduction** (5 minutes) - Ask questions: How do you determine the input that will produce the worst-case scenario for a given algorithm? - What is the worst-case time complexity of the bubble sort algorithm, and how is it derived? - Discuss why in real life, most of the time we do the worst-case analysis of an algorithm. - Solving examples based on Worst-case analysis of an algorithm. 2. **Development** (30 minutes) a. Understand Objectives of Analyzing Algorithms - how the worst, and best case of an algorithm depends on the size of the user input value. - Elaborate why the running time varies among different instances of the input. - Defining Best and Worst Case of an algorithm. - Understanding the examples based on worst-case Analysis b. Calculate the upper limit of the execution time of an algorithm.

	c. Solving Example of Linear Search and analyzing the searching problem. d. Understanding how the worst, case of an algorithm depends on the size of the user input value. e. Best Case and Worst-Case Analysis of an Algorithm Link: https://youtu.be/5g7K86jYto8 f. Summarize the Objectives of Best Case and Worst-Case Analysis of an Algorithm 3. **Exercise** (10 minutes) – - Have a discussion to summarize the lecture - Ask Questions Related to Topics
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: **a. Introduction to Algorithms** by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition. **b. Algorithms** by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition. **Home work**: **Activity** : Solving Examples based on Worst Case Analysis of an Algorithm Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	**1.** Reflective Questions (What, Why, Who?). Allow students to answer and discuss. **2.** Asking open-ended questions on Worst-case analysis through nearpod. **3.** MCQ / Sessional Test / Assignments Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1.5	Course Name: Design and Analysis of Algorithms	Course No.: COM-301

Topics	**Average-case analysis**
Objectives	At the end of the lesson the student shall be able to: a. Understand how the average case of an algorithm depends on the size of the user input value. b. Explain the importance of average case analysis compared to worst-case and best-case analyses. c. perform average case analysis on various algorithmic problems. d. Apply average case analysis to real-world problems and algorithms. e. Understand how input distributions affect average case analysis.
Teaching Aids (if any)	a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	1. **Introduction** (5 minutes) - Ask questions: What mathematical tools and techniques are commonly used in average case analysis? - Can you explain the concept of probability distributions and how they are used in average case analysis? - How does the average case performance of a sorting algorithm differ from its worst-case performance? Use an example like QuickSort or MergeSort. - Solve examples based on Average-case analysis of an algorithm. 2. **Development** (30 minutes) a. Discuss a real-world scenario where average case analysis provides more meaningful insights than worst-case analysis. b. Can you analyze the average case performance of hash table operations such as insertion, deletion, and search? c. Understanding how the average case of an algorithm depends on the size of the user input value. d. Best Case, Worst Case and Average Case Analysis of an Algorithm Link: https://youtu.be/5g7K86jYto8

	e. Summarize the Objectives of Average Case Analysis of an Algorithm 3. **Exercise** (10 minutes) – - Have a discussion to summarize the lecture - Ask Questions Related to Topics
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: **a. Introduction to Algorithms** by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition. **b. Algorithms** by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition. **Homework**: **Activity**: Solving Examples based on Average Case Analysis of an Algorithm Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	**1.** Reflective Questions (What, Why, Who?). Allow students to answer and discuss. **2.** Asking open-ended questions on Average-case analysis through nearpod. **3.** MCQ / Sessional Test / Assignments Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1.6	Course Name: Design and Analysis of Algorithms	Course No.: COM-301

Topics	**Use of order notations and related results**
Objectives	At the end of the lesson, the student shall be able to: a. Understanding the process of approaching a value or curve arbitrarily closely. b. Explain the significance of Resources for an algorithm that is usually expressed as a function regarding input. c. Describe the concept of Asymptotic Notation. d. Understanding the characteristics of an algorithm's efficiency. e. Solve examples on Big-oh, Omega, and Theta Notations.
Teaching Aids (if any)	a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	1. **Introduction** (5 minutes) - Ask questions: What do you understand by running time for an algorithm? How complexity of an algorithm can be analyzed? What is the best way to study Function growth efficiently? - Focus on the significance of Resources for an algorithm. - Have a discussion on the Asymptotic Notation and its applications. - Introducing three notations that are used to calculate the running time complexity of an algorithm 2. **Development** (30 minutes) a. Divide & Conquer Strategy - Explaining which function grows slowly with the input size as compared to others? - Elaborate ways of comparing different running times b. Describe the concept of Asymptotic Notations - Understand the ways to find the approximation in terms of upper, lower and tight bound. - Illustrating asymptotic tight bound with help of examples.

	- Calculating Asymptotic Upper and Lower Bound of a function. c. Describe why the use of O notations is preferable instead of Θ notations even though Θ could be more appropriate. d. Explaining factors that play a significant role in understanding the running time for an algorithm? e. Asymptotic Notations | Big O | Big Omega | Theta Notations Link: https://www.youtube.com/watch?v=7dz8Iaf_weM f. Summarize the Objectives of Use of order notations and related results. 3. **Exercise** (10 minutes) – - Have a discussion to summarize the lecture - Ask Questions Related to Topics
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: **a. Introduction to Algorithms** by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition. **b. Algorithms** by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition. **Home work**: 3. **Activity** : Understanding the Significance of Big-oh , Omega and Theta Notations. **Activity References**: **a.** https://learnxinyminutes.com/docs/asymptotic-notation **b.** https://www.khanacademy.org/computing/computer-science/algorithms/asymptotic-notation/a/asymptotic-notation Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	**1.** Reflective Questions (What, Why, Who?). Allow students to answer and discuss. **2.** Google form Quiz on solving examples of Big-oh , Omega and Theta Notations **3.** MCQ / Sessional Test / Assignments Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1.	Course Name: Design and Analysis of Algorithms	Course No.: COM-301

Topics	Recurrence Equations
Objectives	At the end of the lesson the student shall be able to: a. Define what a recurrence equation is and understand its role in analyzing algorithms. b. Identify the basic components of recurrence equations, including the initial conditions and recurrence relations. c. Explain the significance of recurrence equations in the analysis of recursive algorithms. d. Analyze the efficiency of recursive algorithms by solving their recurrence equations.
Teaching Aids (if any)	a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	1. **Introduction** (5 minutes) - Ask questions: What is a recurrence equation in the context of algorithm analysis? - Why are recurrence equations important in the design and analysis of algorithms? - How do recurrence equations help in understanding the performance of recursive algorithms? - What methods can be used to solve recurrence equations? - How does the substitution method work for solving recurrence equations? 2. **Development** (30 minutes) a. How can recurrence equations be applied to dynamic programming problems? b. What is the difference between homogeneous and non-homogeneous recurrence equations? c. How can you solve non-homogeneous recurrence equations in algorithm analysis?? d. Recurrence Equations

	Link:https://www.youtube.com/watch?v=G5B9on-JUoM&list=PL8tc66sMn9KiZiMN1D5DUue31QYTXVsXj e. Summarize the Objectives of Recurrence Equations. 3. **Exercise** (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: **a. Introduction to Algorithms** by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition. **b. Algorithms** by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition. Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	**1.** Reflective Questions (What, Why, Who?). Allow students to answer and discuss. **2.** Google form Quiz on solving examples of Recurrence Equations **3.** MCQ / Sessional Test / Assignments Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1.8	Course Name: Design and Analysis of Algorithms	Course No.: COM-301

Topics	**Divide and conquer recurrences**
Objectives	At the end of the lesson the student shall be able to: a. Understand the Objectives of Divide-and-conquer algorithms. b. Explaining how Recurrences are used to analyze the computational complexity of an algorithm. c. Describe the process of Dividing the problem into smaller sub-problems. d. Understanding the concept of Solving those sub-problems. e. Combining the solutions for those smaller subproblems to solve the original problem.
Teaching Aids (if any)	a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	1. **Introduction** (5 minutes) - Ask questions: What are the resources for an algorithm that are usually expressed as a function regarding input? Elaborate the significance of Asymptotic Notation. Explain process of approaching a value or curve arbitrarily closely. - Focus on how we can describe a function in terms of its values on smaller inputs. - Have a discussion on Binary search to justify the concept of Divide and conquer recurrences. 2. **Development** (30 minutes) a. Understand Objectives of Learning Algorithms - Divide the problem into subproblems that are smaller instances of the same problem. - Conquer the subproblems by solving them recursively. If the subproblems are small enough, solve them trivially or by "brute force."

	- Combine the subproblem solutions to give a solution to the original problem. - Solving Linear Recurrences b. Understanding how the recursive nature of D&C leads to recurrences. c. Explaining how the Recurrences are used to analyze the computational complexity of divide-and-conquer algorithms. d. Solving the Binary search problem using D & C approach. e. Divide and Conquer Approach Link: https://www.youtube.com/watch?v=I8w2XN0w-fQ f. Summarize the Objectives of Recurrence Relation and How to Write Binary Search using Recurrence Relation. 3. **Exercise** (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: **a.** **Introduction to Algorithms** by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition. **b.** **Algorithms** by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition. **Home work**: 3. **Activity** : Understanding the working of Divide and conquer recurrences **Activity References**: **a.** https://eng.libretexts.org/Bookshelves/Computer_Science/Programming_and_Computation_Fundamentals/Mathematics_for_Computer_Science_(Lehman_Leighton_and_Meyer)/05%3A_Recurrences/21%3A_Recurrences/21.04%3A_Divide-and-Conquer_Recurrences **b.** http://www.cs.cmu.edu/afs/cs/academic/class/15451-s14/www/LectureNotes/lecture1_supplement.pdf Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	**1.** Reflective Questions (What, Why, Who?). Allow students to answer and discuss.

	2. Asking Open ended questions on Objectives of Divide and conquer recurrences **3.** MCQ / Sessional Test / Assignments Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1.9	Course Name: Design and Analysis of Algorithms	Course No.: COM-301

Topics	**Recurrence relations: substitution method**
Objectives	At the end of the lesson the student shall be able to: a. Understand Objectives of condensed way of proving an asymptotic bound on a recurrence by induction. b. Explaining how to find a closed-form bound on the recurrence. c. Conclude that how substitution method is a powerful approach that is able to prove upper bounds for almost all recurrences. d. Understanding the use of induction in solving recurrence relations. e. Describing importance of adding additional terms to the upper bound. f. Solve Substitution method examples.
Teaching Aids (if any)	a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	1. **Introduction** (5 minutes) - Ask questions: How the recursive nature of D&C leads to recurrences. How the Recurrences are used to analyze the computational complexity of divide-and-conquer algorithms. Explain the process of Solving Linear Recurrences. - Focus on obtaining a function defined on the natural numbers that satisfies the recurrence. - Have a discussion on Using the mathematical induction to find the boundary condition and show that the guess is correct. 2. **Development** (30 minutes) a. Understand Objectives of the condensed way of proving an asymptotic bound on a recurrence by induction. - Process to get a recurrence relation for time complexity - Get the running time on an input of size n as a function of n and the running time on inputs of smaller sizes. - Guess for the solution and then use mathematical induction to prove the guess is correct or incorrect.

	- Determine a tight asymptotic lower bound for the example recurrence relation. b. Describing the importance of adding additional terms to the upper bound. c. Conclude that how substitution method is a powerful approach that is able to prove upper bounds for almost all recurrences. d. Substitution Method Link : https://youtu.be/icS-e8RaCyo e. Summarize the Objectives of learning Recurrence relations: substitution method 3. **Exercise** (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: **a. Introduction to Algorithms** by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition. **b. Algorithms** by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition. **Home work**: 3. **Activity** : Understanding the working of Substitution Method **Activity References**: **a.** https://www.cs.cornell.edu/courses/cs3110/2014sp/recitations/24/using-the-substitution-and-master-method.html **b.** https://walkccc.me/CLRS/Chap04/4.3/ Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	**1.** Reflective Questions (What, Why, Who?). Allow students to answer and discuss. **2.** Nearpod activity on solving given example on substitution method **3.** MCQ / Sessional Test / Assignments Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1.10	Course Name: Design and Analysis of Algorithms	Course No.: COM-301

Topics	Recurrence trees method
Objectives	At the end of the lesson, the student shall be able to: a. Understand the Objectives of drawing a recurrence tree and calculate the time taken by every level of the tree. b. Explaining how master theorem allows many recurrence relations of a form to be converted to Θ-notation directly, without doing an expansion of the recursive relation. c. Conclude how the master theorem always yields asymptotically tight bounds to recurrences from divide and conquer algorithms d. Understanding the way how recursion trees can be useful for gaining intuition about the closed form of a recurrence.
Teaching Aids (if any)	a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	1. **Introduction** (5 minutes) - Ask questions: - What is a recurrence tree, and how does it help in solving recurrence relations? - How do you construct a recurrence tree for a given recurrence relation? - What are the key components of a recurrence tree (e.g., root, nodes, levels)? - How do you determine the number of levels in a recurrence tree? 2. **Development** (30 minutes) a. Construct a recurrence tree for the recurrence relation T(n)=2T(n/2)+nT(n) = 2T(n/2) + nT(n)=2T(n/2)+n and solve it. b. How would you solve the recurrence relation T(n)=T(n−1)+nT(n) = T(n-1) + nT(n)=T(n−1)+n using a recurrence tree? c. Provide an example of a more complex recurrence relation and demonstrate how to solve it using a recurrence tree. d.Recursive Tree Method Link : https://www.youtube.com/watch?v=HBUFYO9gqvA

	Summarize the Objectives of learning Recurrence relations: Recursive Tree Method 3. **Exercise** (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: **a. Introduction to Algorithms** by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition. **b. Algorithms** by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition. **Home work**: 3. **Activity** : Understanding the working of Recursive Tree Method **Activity References**: **a.** https://www.cs.cornell.edu/courses/cs3110/2012sp/lectures/lec20-master/lec20.html Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	**1.** Reflective Questions (What, Why, Who?). Allow students to answer and discuss. **2.** Google form Quiz on solving given examples on Recursive Tree Method. **3.** MCQ / Sessional Test / Assignments Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1.11	Course Name: Design and Analysis of Algorithms	Course No.: COM-301

Topics	Master's theorem
Objectives	At the end of the lesson the student shall be able to: a. Classify the function f(n) and apply the appropriate case of Master's Theorem to determine the asymptotic behavior of the recurrence relation. b. Apply Master's Theorem to analyze the time complexity of real-world algorithms, such as Merge Sort, Binary Search, and Strassen's Algorithm for matrix multiplication. c. Conclude how the master theorem always yields asymptotically tight bounds to recurrences from divide and conquer algorithms. d. Identify recurrence relations of the form T(n)=aT(n/b)+f(n) and recognize when Master's Theorem is applicable.
Teaching Aids (if any)	a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	1. **Introduction** (5 minutes) - Ask questions: What is the Master's Theorem and in what context is it used in the design and analysis of algorithms? - Why is the Master's Theorem important for analyzing the time complexity of recursive algorithms? - What are the three cases of the Master's Theorem, and what conditions determine which case to use? - What is the standard form of a recurrence relation to which the Master's Theorem can be applied? - What are the three cases of the Master's Theorem, and what conditions determine which case to use? 2. **Development** (30 minutes) a. How do you identify the values of a, b, and f(n) in a recurrence relation for the application of the Master's Theorem? b. Can you apply the Master's Theorem to the recurrence T(n)=2T(n/2)+ n? Explain your steps and the result. c. Apply the Master's Theorem to the recurrence $T(n)=4T(n/2)+n^2$ and explain the result.

	d. Master Theorem Link : https://youtu.be/vny0ZVVLpbA 3. Summarize the Objectives of learning Recurrence relations: Master Theorem 4. **Exercise** (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: **a. Introduction to Algorithms** by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition. **b. Algorithms** by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition. **Home work**: 3. **Activity** : Understanding the working of Master Theorem **Activity References**: **a.** https://www.programiz.com/dsa/master-theorem **b.** https://www.cs.cornell.edu/courses/cs3110/2012sp/lectures/lec20-master/lec20.html Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	**1.** Reflective Questions (What, Why, Who?). Allow students to answer and discuss. **2.** Google form Quiz on solving given examples on Master Theorem **3.** MCQ / Sessional Test / Assignments Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 1.12	Course Name: Design and Analysis of Algorithms	Course No.: COM-301

Topics	**Applications of Master's Theorem**
Objectives	At the end of the lesson, the student shall be able to: a. Apply the Master's Theorem to solve recurrence relations commonly encountered in algorithm design. b. Recognize recurrence relations that can be solved using the Master's Theorem. c. Use the Master's Theorem to determine the time complexity of various divide-and-conquer algorithms. d. Formulate recurrence relations for real-world problems and solve them using the Master's Theorem.
Teaching Aids (if any)	a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	1. **Introduction** (5 minutes) - Ask questions: - How can Master's Theorem be used to analyze the time complexity of divide-and-conquer algorithms? - Can you provide a step-by-step example of applying Master's Theorem to a specific recurrence relation? - How does the theorem help simplify the process of finding the asymptotic complexity of recursive algorithms? - What are the conditions for each case of Master's Theorem? 2. **Development** (30 minutes) a. How does Master's Theorem relate to the recursion tree method and the iteration method? b. Can you derive the Master's Theorem from first principles and explain the intuition behind it? c. Explore the relationship between Master's Theorem and other methods for solving recurrence relations. d. Master Theorem Link : https://www.youtube.com/watch?v=4aePc0qOL70

	e. Summarize the Objectives of learning Application of Master Theorem 3. **Exercise** (10 minutes) – - Have a discussion to summarize the lecture - Ask Questions Related to Topics
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: **a. Introduction to Algorithms** by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition. **b. Algorithms** by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition. **Home work**: 3. **Activity** : Understanding the working of Application of Master Theorem **Activity References**: **a.** https://www.programiz.com/dsa/master-theorem **b.** https://www.cs.cornell.edu/courses/cs3110/2012sp/lectures/lec20-master/lec20.html Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	**1.** Reflective Questions (What, Why, Who?). Allow students to answer and discuss. **2.** Google form Quiz on solving given examples on Application of Master Theorem **3.** MCQ / Sessional Test / Assignments Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2.1	Course Name: Design and Analysis of Algorithms	Course No.: COM-301

Topics	**Quick Sort and its analysis**
Objectives	At the end of the lesson the student shall be able to: a. Understand how Quick Sort uses a divide and conquer paradigm for sorting. b. Explaining execution trace of partition algorithm. c. Analyse the algorithm and doing its Worst-Case Analysis. d. Elaborating the Relational Formula for Worst Case e. Deriving Relational Formula for Randomized Quick Sort f. Doing Best and Average Case Analysis
Teaching Aids (if any)	a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	1. **Introduction** (5 minutes) - Ask questions: How to calculate the time taken by every level of a recursion tree? How master theorem allows many recurrence relations of a form to be converted to Θ-notation directly? What is the main principal behind working of Divide & Conquer strategy? - Understand how quick sort consumes relatively fewer resources during execution. 2. **Development** (30 minutes) a. Discuss the Pseudo Code for recursive QuickSort function b. Explain the Pseudo code for partition () function. c. Understand how quick sort uses only a small auxiliary stack d. Elaborate the procedure of Partitioning the Array e. Understand about extremely short inner loop of algorithm. f. Performing a thorough mathematical analysis of the algorithm. g. Listing the advantages and disadvantages of Quick sort Algorithm. h. Analyzing the Worst Case, Best Case and Average Case Complexity of the algorithm.

	i. How to optimize QuickSort so that it takes O(Log n) extra space in worst case? j. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/ Quick Sort Algorithm - Divide and Conquer Link : https://youtu.be/IBnyUwCoY4I How Quick Sort Works | Performance of Quick Sort with Example Link: https://youtu.be/tWCaFVJMUi8 k. Summarize the Objectives of learning Quick Sort Algorithm and performing its analysis 3. **Exercise** (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: **a.** **Introduction to Algorithms** by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition. **b.** **Algorithms** by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition. **Home work**: 3. **Activity** : Understanding the working of Quick Sort Algorithm and performing its analysis **Activity References**: **a.** https://www.gatevidyalay.com/tag/quick-sort-in-daa/ **b.** https://www.codesdope.com/course/algorithms-quicksort/ Spend 5 minutes to wrap up and consolidate the leanings.

Evaluation	**1.** Reflective Questions (What, Why, Who?). Allow students to answer and discuss. **2.** Nearpod / Mentimeter short Quiz on Quick Sort Algorithm and its analysis **3.** MCQ / Sessional Test / Assignments Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2.2	Course Name: Design and Analysis of Algorithms	Course No.: COM-301

Topics	**Merge Sort recurrence**
Objectives	At the end of the lesson the student shall be able to: a. Understand how Merge Sort uses a divide and conquer paradigm for sorting. b. Explaining how algorithm divides the problem into sub problems and solves them individually. c. Elaborating the process of combining the results of sub problems to get the solution of the original problem d. Analyse the Time Complexity of Merge Sort(Best, Average and Worst Case). e. Explaining Merge Sort Applications.
Teaching Aids (if any)	a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	1. **Introduction** (5 minutes) - Ask questions: How Quick Sort uses a divide and conquer paradigm for sorting? What is Worst-Case Complexity of Quick Sort. How to derive the Relational Formula for Randomized Quick Sort? - Explaining the working principal of Merge Sort using suitable example. 2. **Development** (30 minutes) a. Discuss the Top-down Merge Sort Implementation and see how Merge sort repeatedly breaks down a list into several sub lists until each sub list consists of a single element. b. Consider an example to understand the approach better. c. Understand the process of Merging of two lists. d. Elaborating the Implementation of Merge Sort. e. Explaining the process how Bottom-Up merge sort approach uses iterative methodology. f. Analyzing the Time Complexity of Merge Sort recurrence.

	g. Explaining how Merge Sort is useful for sorting linked lists in O(nlogn) time. h. Listing the reasons behind the Slower processing comparative to the other sort algorithms for smaller tasks. i. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/ Recursion tree method: intuition | Merge Sort Link : https://youtu.be/C4JjXc0htp0 Solving the Merge Sort recurrence Link: https://youtu.be/LPfHV3xWwXg j. Summarize the Objectives of learning Merge Sort Algorithm and performing its analysis 3. **Exercise** (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: **a. Introduction to Algorithms** by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition. **b. Algorithms** by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition. **Home work**: 3. **Activity** : Understanding the working of Merge Sort Algorithm and performing its analysis **Activity References**: **a.** https://www.gatevidyalay.com/tag/merge-sort-algorithm-in-daa/ **b.** https://www.interviewbit.com/tutorial/merge-sort-algorithm/ Spend 5 minutes to wrap up and consolidate the leanings.

Evaluation	**1.** Reflective Questions (What, Why, Who?). Allow students to answer and discuss. **2.** Mentimeter short Quiz on Merge Sort Algorithm and its analysis **3.** MCQ / Sessional Test / Assignments Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2.3	Course Name: Design and Analysis of Algorithms	Course No.: COM-301

Topics	**Strassen's matrix multiplication**
Objectives	At the end of the lesson the student shall be able to: a. Understand the significance of Strassen's Algorithm for matrix multiplication. b. Explaining how this algorithm is faster than the naive matrix multiplication algorithm. c. Elaborating the conditions for Strassen's algorithm to work. d. Understanding the Procedure of Strassen matrix multiplication. e. Analyse the Time Complexity of Strassen's matrix multiplication. f. Explaining the Disadvantages of Strassen's matrix multiplication
Teaching Aids (if any)	a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	1. **Introduction** (5 minutes) - Ask questions: How Merge Sort uses a divide and conquer paradigm for sorting? What are the Applications of Merge Sort ? What is the Time Complexity of Merge Sort algorithm. How Merge Sort uses a divide and conquer paradigm for sorting. - Explaining the basic working principal of Strassen's matrix multiplication. 2. **Development** (30 minutes) a. Discuss the Problem Statement of Naive method during matrix multiplication. b. Analyzing the Complexity for Naïve method of matrix multiplication. c. Understanding the idea of Strassen's method to reduce the number of recursive calls. d. Understanding the formula behind the working of algorithm. e. Calculate the product of A and B (matrix C) with the help of learned formulas. f. Describing the different conditions for Strassen's algorithm to work.

	g. Understanding how Strassen's matrix multiplication algorithm is asymptotically faster than the naive algorithm. h. Explaining the reasons why generally Strassen's Method is not preferred for practical applications. i. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/ Strassen's Matrix Multiplication Link : https://youtu.be/Te08SanNEN8 Strassen's Matrix Multiplication - Divide and Conquer - Analysis of Algorithm Link: https://youtu.be/UnpySHwAJsQ j. Summarize the Objectives of learning Strassen's matrix multiplication and performing its analysis 3. **Exercise** (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: **a. Introduction to Algorithms** by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition. **b. Algorithms** by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition. **Home work**: 3. **Activity** : Understanding the working of Strassen's matrix multiplication and performing its analysis **Activity References**: **a.** https://iq.opengenus.org/strassens-matrix-multiplication-algorithm/ **b.** https://medium.com/swlh/strassens-matrix-multiplication-algorithm-936f42c2b344 Spend 5 minutes to wrap up and consolidate the leanings.

Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Google form short Quiz on Strassen's Matrix Multiplication and its analysis 3. MCQ / Sessional Test / Assignments Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2.4	Course Name: Design and Analysis of Algorithms	Course No.: COM-301

Topics	**Fast multiplication of large integers**
Objectives	At the end of the lesson the student shall be able to: a. Analysing asymptotic efficiency by ingenious application of the divide-and-conquer technique. b. Explaining efficient manipulation of large integers. c. Elaborating the basic idea of the algorithm. d. Understanding the Procedure Fast multiplication e. Analyse the Complexity of Fast multiplication and its optimization.
Teaching Aids (if any)	a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	1. **Introduction** (5 minutes) - Ask questions: What is the significance of Strassen's Algorithm for matrix multiplication? What are the conditions for Strassen's algorithm to work. How Strassen's matrix multiplication algorithm is asymptotically faster than the naive algorithm? why generally Strassen's Method is not preferred for practical applications? - Understanding how much effort does it take to do long multiplication of two numbers. 2. **Development** (30 minutes) a. Discuss the Problem Statement of Naive method during matrix multiplication. b. Analyzing the process of addition of long numbers. c. Understanding the idea of short multiplication: a number times a digit. d. Understanding the formula behind the working of algorithm. e. Perform the analysis of long multiplication.

	f. Explaining Karatsuba's method for multiplying numbers of any length g. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/ Large Integer Multiplication - Divide and Conquer - Analysis of Algorithm Link : https://youtu.be/BwHZXuzux44 Multiplying Long Integers Using Divide and Conquer Technique Link: https://youtu.be/_YIAcrBwpas h. Summarize the Objectives of learning Multiplication of Long Integers (Faster than Long Multiplication) 3. **Exercise** (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: **a. Introduction to Algorithms** by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition. **b. Algorithms** by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition. **Home work**: 3. **Activity** : Understanding the working of Multiplication of Long Integers (Faster than Long Multiplication) **Activity References**: **a.** https://people.mpiinf.mpg.de/~mehlhorn/ftp/chapter2A-en.pdf **b.** https://www.brainkart.com/article/Multiplication-of-Large-Integers_8025/ Spend 5 minutes to wrap up and consolidate the leanings.

Evaluation	1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Google form short Quiz on Multiplication of Long Integers (Faster than Long Multiplication) 3. MCQ / Sessional Test / Assignments Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2.5	Course Name: Design and Analysis of Algorithms	Course No.: COM-301

Topics	**Binary Search trees**
Objectives	At the end of the lesson the student shall be able to: a. Understanding the nodes arrangement in a Binary Search Tree. b. Explaining the basic operations associated with a tree. c. Implementation of Search, Delete and Insert operations in a BST. d. Understanding the architecture of a basic binary search algorithm. e. Analyse the Attributes of Binary Search Tree. f. Explaining the Types of Binary Trees g. Practice Problems Based on Binary Search Trees
Teaching Aids (if any)	a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	1. **Introduction** (5 minutes) - Ask questions: What do you understand by a tree? What is the difference between a tree and a graph? How to Perform the analysis of long multiplication? How we can analyze asymptotic efficiency by ingenious application of the divide-and-conquer technique? - Understanding the importance of Traversal in Binary Search Trees. 2. **Development** (30 minutes) a. Discuss how we can recursively define a rooted binary tree. b. Explaining In-Order, PREORDER and POSTORDER Tree Walk traversals along with suitable examples. c. Understanding the idea of Querying a Binary Search Tree. d. Understanding the process of Tree Search and Iterative Tree Search. e. How we can define Successor and predecessor in a BST. f. Explaining algorithms for Tree Insertion and Deletion. g. Practice Problems Based on Binary Search Trees h. Explaining How Good Are Binary Search Trees? i. Understanding the concept of Balanced Search Trees.

	j. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/ Binary Search Tree | BST | Design & Algorithms Link : https://youtu.be/cvooTtquDlQ Insertion and Traversal in BST Link: https://youtu.be/sXABdGalFNg k. Summarize the Objectives of learning Binary Search trees and doing analysis. 3. **Exercise** (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: **a. Introduction to Algorithms** by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition. **b. Algorithms** by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition. **Home work**: 3. **Activity** : Understanding the working of Binary Search trees **Activity References**: **a.** https://www.gatevidyalay.com/binary-search-trees-data-structures/ **b.** https://www.studytonight.com/data-structures/binary-search-tree Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	**1.** Reflective Questions (What, Why, Who?). Allow students to answer and discuss. **2.** Nearpod Activity on problems based on Binary Search trees. **3.** MCQ / Sessional Test / Assignments

	Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2.6	Course Name: Design and Analysis of Algorithms	Course No.: COM-301

Topics	**Priority queues**
Objectives	At the end of the lesson the student shall be able to: a. Introduction to the concept of Priority queues b. Explaining the basic operations associated with a Priority queue. c. Understanding how to change priority of an arbitrary element. d. How to join two Priority queues. e. Explaining the basic operations associated with a Max Priority queue. f. Implementation of Priority queues with various techniques.
Teaching Aids (if any)	a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	1. **Introduction** (5 minutes) - Ask questions: What are the Attributes of Binary Search Tree? How we can recursively define a rooted binary tree? How we can define Successor and predecessor in a BST. What do you understand by Balanced Search Trees? - Understanding how the higher priority elements are served first in a priority queue. 2. **Development** (30 minutes) a. Understanding the process of Assigning Priority Value in a priority queue. b. Explaining Difference between Priority Queue and Normal Queue. c. Understanding the Implementation of Priority Queues using different Data Structures. d. Explaining the process of Inserting an Element into the Priority Queue. e. Describing the process of Deleting an Element from the Priority Queue.

	f. Understanding concept of Peeking from the Priority Queue (Find max/min). g. Explaining concept of Extract-Max/Min from the Priority Queue. h. Explaining the Priority Queue Applications. i. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/ Priority Queue Data Structure Link : https://www.youtube.com/watch?v=NlEwbC6Nt0c Priority Queue - Implementation with Example Link: https://youtu.be/OxhYCLWMdHs j. Summarize the Objectives of learning Binary Search trees and doing analysis. 3. **Exercise** (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: **a. Introduction to Algorithms** by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition. **b. Algorithms** by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition. **Home work**: 3. **Activity** : Understanding the working of Priority queues **Activity References**: **a.** http://www.infocobuild.com/education/audio-video-courses/computerscience/DesignAnalysisOfAlgorithms-CMI/lecture-34.html **b.** https://algs4.cs.princeton.edu/24pq/ Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	**1.** Reflective Questions (What, Why, Who?). Allow students to answer and discuss.

	2. Nearpod Activity on problems based on Priority queues **3.** MCQ / Sessional Test / Assignments Spend 5 minutes to evaluate student assimilation of the lesson contents

Lesson Plan No. 2.7	Course Name: Design and Analysis of Algorithms	Course No.: COM-301

Topics	**Heaps, Heap Sort and its analysis**
Objectives	At the end of the lesson the student shall be able to: a. Utilize the Heap to sort the given array. b. Understanding special heap properties. c. Describe the concept of Complete and incomplete Binary tree. d. Create a Heap data structure from unsorted list. e. Explaining the basic operations associated with Min- Heap and Max Heap. f. Analyse the Complexity of Heap Sort. g. Implementing Heap Sort Algorithm.
Teaching Aids (if any)	a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	1. **Introduction** (5 minutes) - Ask questions: What are the basic operations associated with a Priority queue? Explain the different applications of Priority Queue. How we can assign Priority Value in a priority queue. How the higher priority elements are served first in a priority queue. How we can define Successor and predecessor in a BST. - Understanding how the heap is also the data structure that serves as a cornerstone of a theoretically important sorting algorithm called heapsort. 2. **Development** (30 minutes) a. Understanding the conditions required to form a heap. b. To check whether the binary tree is essentially complete. c. Understanding the process of implementing heap property. d. Explaining the process of Inserting an Element into the Priority Queue. e. Describing the list of important properties of heaps.

	f. Understanding concept of Heap and its array representation. g. How can we construct a heap for a given list of keys? h. Implementation of Heap Sort Algorithm using heapify function. i. Analyzing the Worst Case , Best Case and Average Case Complexity of an algorithm. j. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/ Heap Sort | Heapify Method | Build Max Heap Algorithm Link : https://youtu.be/Q_eia3jC9Ts Heap sort with Example | Heapify Method Link: https://youtu.be/nJ6FdAIr_6g k. Summarize the Objectives of learning Heaps, Heap Sort and performing its analysis 3. **Exercise** (10 minutes) – - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: **a. Introduction to Algorithms** by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition. **b. Algorithms** by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition. **Home work**: 3. **Activity** : Practice to implement Heap Sort and learn its analysis **Activity References**: **a.** https://www.kodnest.com/free-online-courses/algorithm-2/lessons/transform-and-conquer-approach/topic/heaps-and-heap-sort/ **b.** https://www.interviewbit.com/tutorial/heap-sort-algorithm/ Spend 5 minutes to wrap up and consolidate the leanings.
Evaluation	**1.** Reflective Questions (What, Why, Who?). Allow students to answer and discuss.

	2. Nearpod quiz Activity on problems based on Heap Sort and its analysis 3. MCQ / Sessional Test / Assignments Spend 5 minutes to evaluate student assimilation of the lesson contents