



Kot Bhalwal, Jammu



Model Institute of Engineering
& Technology (Autonomous)
Dr. Arun K. Gupta Teaching-Learning Centre

Department of Computer Applications

Details of Lesson Plan

S.No.	Particulars	Details
1.	Course Name	Algorithm Design & Analysis
2.	Course Code	PSCSATC-224
3.	Academic Year	2023-24
4.	Semester	2 nd
5.	Number of Lesson plans	40
6.	Faculty Assigned	Saurabh Sharma

Faculty Signature

Lesson Plan No. 1.1	Course Name: Algorithm Design & Analysis Topic Name: Mathematical Preliminaries	Course No.: PSCSATC-224
----------------------------	--	--------------------------------

Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand Objectives of Learning Algorithms Explain significance of Preliminary Terms Describe Mathematical Model of a Computer Understanding the concept of Demonstrating Incorrectness. Reasoning about Correctness of an algorithm.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> Ask questions: What do you understand by an algorithm? Define the term problem instance? Does 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. Development (30 minutes) <ol style="list-style-type: none"> Understand Objectives of Learning Algorithms <ul style="list-style-type: none"> 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. Describe Mathematical Model of a Computer <ul style="list-style-type: none"> Explain the significance of RAM Model and It working. Working Principles of EM Model. PRAM Model working criteria along with advantages and application areas. Expressing Algorithms <ul style="list-style-type: none"> 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



	<p>d. NPTEL Link: Introduction and Mathematical Preliminaries https://nptel.ac.in/courses/106/106/106106131/</p> <p>Introduction to Fundamental mathematical concepts Link: https://youtu.be/LzYoAC0IxKI</p> <p>RAM Model of Computation Link: https://youtu.be/xbVRHTGor70</p> <p>e. Summarize the Objectives of Learning Algorithms and Working of various Mathematical Models of a Computer.</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	<p>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</p> <p>2. Suggested Reading books:</p> <ol style="list-style-type: none"> 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. <p>Home work:</p> <p>3. Activity : Understanding the working of Mathematical Model of a Computer.</p> <p>Activity References:</p> <ol style="list-style-type: none"> 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 <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<ol style="list-style-type: none"> 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



Kot Bhalwal, Jammu

	Spend 5 minutes to evaluate student assimilation of the lesson contents
--	---



Kot Bhalwal, Jammu



Lesson Plan No. 1.2	Course Name: Algorithm Design & Analysis Topic Name: Time complexity and Space complexity	Course No.: PSCSATC-224
----------------------------	--	--------------------------------

Objectives	At the end of the lesson the student shall be able to: a. Understanding Objectives of micro analysis and the macro analysis. b. Explain significance of determining the order of the magnitude of the statement. c. Describe the concept of Posteriori Analysis. d. Understanding the time-space trade-off involving a problem e. Solve examples on finding Time and 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. 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 solve with different methods. b. 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. c. Describing what it Takes to Create a Good Algorithm?



	<p>d. Explaining factors that play a significant role in the long-term usage of an algorithm.</p> <p>e. NPTEL Link: Time and Space Trade-off https://nptel.ac.in/courses/106/106/106106131/</p> <p>Space and Time Complexity of an Algorithm Link: https://www.youtube.com/watch?v=Si9MzFqBs8E</p> <p>Time Complexity in Analysis of Algorithm Link: https://www.youtube.com/watch?v=88Rq4U2mioQ</p> <p>f. Summarize the Objectives of Learning Algorithms and Working of various Mathematical Models of a Computer.</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none">- Have discussion to summarize the lecture- Ask Questions Related to Topics
<p>Closure</p>	<p>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</p> <p>2. Suggested Reading books:</p> <ul style="list-style-type: none">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. <p>Home work:</p> <p>3. Activity : Understanding the Significance of Time and Space Complexity of an algorithm.</p> <p>Activity References:</p> <ul style="list-style-type: none">a. https://towardsdatascience.com/space-and-time-complexity-in-computer-algorithms-a7ffe9e4683b. https://www.hackerearth.com/practice/basic-programming/complexity-analysis/time-and-space-complexity/tutorial/ <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>



Kot Bhalwal, Jammu

Model Institute of Engineering & Technology (Autonomous) Lesson Plan

Evaluation	<ol style="list-style-type: none">1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.2. Google form Quiz on the topic Time and Space Complexity of an algorithm.3. MCQ / Sessional Test / Assignments <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>
-------------------	---



Dr. Arun K. Gupta Teaching-Learning Centre

Version 1.1

श्रेष्ठ

श्रम

नवीनता

Please Do Not Print Unless Necessary



Lesson Plan No. 1.3	Course Name: Algorithm Design & Analysis Topic Name: Worst-case and Average-case analysis	Course No.: PSCSATC-224
----------------------------	--	--------------------------------

Objectives	At the end of the lesson, the student shall be able to: a. Understand how the worst, average, and best 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 Average 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: What do you understand Time Complexity of an algorithm? - How the space complexity of any algorithm is calculated? - Have a discussion on why in real life, most of the time we do the worst-case analysis of an algorithm. - Solving examples based on Worst-case and Average-case analysis of an algorithm. 2. Development (30 minutes) a. Understand Objectives of Analyzing Algorithms - how the worst, average, 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, Average and Worst Case of an algorithm. - Understanding the examples based on Best Case and 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.



	<p>d. Understanding how the worst, average, and best case of an algorithm depends on the size of the user input value.</p> <p>e. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/</p> <p>Best Case, Worst Case and Average Case Analysis of an Algorithm Link: https://youtu.be/5g7K86jYto8</p> <p>Analysis of Algorithms Link: https://youtu.be/A_Y4bMFp3bo</p> <p>f. Summarize the Objectives of Best Case, Worst Case and Average Case Analysis of an Algorithm</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
<p>Closure</p>	<p>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</p> <p>2. Suggested Reading books:</p> <ul style="list-style-type: none"> 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. <p>Home work: Activity : Solving Examples based on Best Case, Worst Case and Average Case Analysis of an Algorithm</p> <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
<p>Evaluation</p>	<ol style="list-style-type: none"> 1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Asking open ended questions on Worst-case and Average-case analysis through nearpod. 3. MCQ / Sessional Test / Assignments



Model Institute of Engineering
& Technology (Autonomous)
Lesson Plan

Kot Bhalwal, Jammu

Spend 5 minutes to evaluate student assimilation of the lesson contents



Dr. Arun K. Gupta Teaching-Learning Centre

Version 1.1

श्रेष्ठ

श्रम

नवीनता

Please Do Not Print Unless Necessary



Lesson Plan No. 1.4	Course Name: Algorithm Design & Analysis Topic Name: Use of order notations and related results	Course No.: PSCSATC-224
----------------------------	--	--------------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understanding the process of approaching a value or curve arbitrarily closely. Explain significance of Resources for an algorithm that are usually expressed as a function regarding input. Describe the concept of Asymptotic Notation. Understanding the characteristics of an algorithm's efficiency. Solve examples on Big-oh , Omega and Theta Notations.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> 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 their applications. Introducing three notations that are used to calculate the running time complexity of an algorithm Development (30 minutes) <ol style="list-style-type: none"> Divide & Conquer Strategy <ul style="list-style-type: none"> Explaining which function grows slowly with the input size as compared to others? Elaborate ways of comparing different running times Describe the concept of Asymptotic Notations <ul style="list-style-type: none"> 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.



	<p>c. Describing why the use of O notations is preferable instead of Θ notations even though Θ could be more appropriate.</p> <p>d. Explaining factors that play a significant role in understanding the running time for an algorithm?</p> <p>e. NPTEL Link: Asymptotic Notation and their use https://nptel.ac.in/courses/106/106/106106131/</p> <p>Asymptotic Notations Big O Big Omega Theta Notations Link: https://www.youtube.com/watch?v=7dz8Iaf_weM</p> <p>Understanding growth of common running times of any algorithm Link: https://www.youtube.com/watch?v=1OTX-WXQHCQ</p> <p>f. Summarize the Objectives of Use of order notations and related results.</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none">- Have discussion to summarize the lecture- Ask Questions Related to Topics
Closure	<p>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</p> <p>2. Suggested Reading books:</p> <ul style="list-style-type: none">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. <p>Home work:</p> <p>3. Activity : Understanding the Significance of Big-oh , Omega and Theta Notations.</p> <p>Activity References:</p> <ul style="list-style-type: none">a. https://learnxinyminutes.com/docs/asymptotic-notationb. https://www.khanacademy.org/computing/computer-science/algorithms/asymptotic-notation/a/asymptotic-notation <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<p>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</p>

	<ol style="list-style-type: none">2. Google form Quiz on solving examples of Big-oh , Omega and Theta Notations3. MCQ / Sessional Test / Assignments <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>
--	--



Kot Bhalwal, Jammu

Lesson Plan No. 1.5	Course Name: Algorithm Design & Analysis Topic Name: Divide and conquer recurrences	Course No.: PSCSATC-224
----------------------------	--	--

Objectives	At the end of the lesson the student shall be able to: a. Understand Objectives of Divide-and-conquer algorithms. b. Explaining how Recurrences are used to analyse 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





	<p>b. Understanding how the recursive nature of D&C leads to recurrences.</p> <p>c. Explaining how the Recurrences are used to analyze the computational complexity of divide-and-conquer algorithms.</p> <p>d. Solving the Binary search problem using D & C approach.</p> <p>e. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/</p> <p>Divide and Conquer Approach Link: https://www.youtube.com/watch?v=I8w2XN0w-fQ</p> <p>What is Recurrence Relation How to Write Binary Search Recurrence Relation Link: https://youtu.be/NW-nasIChdo</p> <p>f. Summarize the Objectives of Recurrence Relation and How to Write Binary Search using Recurrence Relation.</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none">- Have discussion to summarize the lecture- Ask Questions Related to Topics
<p>Closure</p>	<p>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</p> <p>2. Suggested Reading books:</p> <ul style="list-style-type: none">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. <p>Home work:</p> <p>3. Activity : Understanding the working of Divide and conquer recurrences</p> <p>Activity References:</p> <ul style="list-style-type: none">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_Recurrencesb. 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	<ol style="list-style-type: none"> 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 <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>

Lesson Plan No. 1.6	Course Name: Algorithm Design & Analysis Topic Name: Recurrence relations: substitution method	Course No.: PSCSATC-224
----------------------------	---	--

Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand Objectives of condensed way of proving an asymptotic bound on a recurrence by induction. Explaining how to find a closed-form bound on the recurrence. Conclude that how substitution method is a powerful approach that is able to prove upper bounds for almost all recurrences. Understanding the use of induction in solving recurrence relations. Describing importance of adding additional terms to the upper bound. Solve Substitution method examples
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> Ask questions: <ul style="list-style-type: none"> 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 shows that the guess is correct. Development (30 minutes) <ol style="list-style-type: none"> Understand Objectives of condensed way of proving an asymptotic bound on a recurrence by induction. <ul style="list-style-type: none"> Process to get a recurrence relation for time complexity Get 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.



	<p>b. Describing importance of adding additional terms to the upper bound.</p> <p>c. Conclude that how substitution method is a powerful approach that is able to prove upper bounds for almost all recurrences</p> <p>d. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/</p> <p>Substitution Method Link : https://youtu.be/icS-e8RaCyo</p> <p>Substitution method Solving Recurrences Link: https://youtu.be/LC96q_6r3BQ</p> <p>e. Summarize the Objectives of learning Recurrence relations: substitution method</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	<p>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</p> <p>2. Suggested Reading books:</p> <ul style="list-style-type: none"> 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. <p>Home work:</p> <p>3. Activity : Understanding the working of Substitution Method</p> <p>Activity References:</p> <ul style="list-style-type: none"> 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/ <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<p>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</p> <p>2. Nearpod activity on solving given example on substitution method</p>



Kot Bhalwal, Jammu

3. MCQ / Sessional Test / Assignments

Spend 5 minutes to evaluate student assimilation of the lesson contents



Lesson Plan No. 1.7	Course Name: Algorithm Design & Analysis Topic Name: Recurrence trees method, Master's theorem and its applications	Course No.: PSCSATC-224
----------------------------	--	--

Objectives	At the end of the lesson the student shall be able to: a. Understand Objectives of draw a recurrence tree and calculate the time taken by every level of 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 that 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: Define the role of the use of induction in solving recurrence relations. Describe the importance of adding additional terms to the upper bound. How substitution method is a powerful approach that is able to prove upper bounds for almost all recurrences. - Have a discussion on the cases used in solving different examples on Masters Theorem. 2. Development (30 minutes) a. Understand the use of recursion tree in visualizing the recurrences. - Process to get the depth of the tree by looking at examples. - Get the amount of work done at each level. - Guess the intuition about the closed form of a recurrence. - prove by induction that your guess is indeed a solution. - Describing importance of adding additional terms to the upper bound.



	<p>b. Conclude that how the asymptotic growth of $f(n)$ is compared to the asymptotic growth of the number of leaves.</p> <p>c. Verifying the subproblem size to identify the appropriate case of Masters Theorem.</p> <p>d. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/</p> <p>Recursive Tree Method & Master Theorem Link : https://youtu.be/vny0ZVVLpbA</p> <p>Master Theorem Link: https://youtu.be/FBKjvXGGCJM</p> <p>e. Summarize the Objectives of learning Recurrence relations: Recursive Tree Method & Master Theorem</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none">- Have discussion to summarize the lecture- Ask Questions Related to Topics
Closure	<p>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</p> <p>2. Suggested Reading books:</p> <ul style="list-style-type: none">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. <p>Home work:</p> <p>3. Activity : Understanding the working of Recursive Tree Method & Master Theorem</p> <p>Activity References:</p> <ul style="list-style-type: none">a. https://www.programiz.com/dsa/master-theoremb. https://www.cs.cornell.edu/courses/cs3110/2012sp/lectures/lec20-master/lec20.html <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<p>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</p>

	<ol style="list-style-type: none">2. Google form Quiz on solving given examples on Recursive Tree Method & Master Theorem3. MCQ / Sessional Test / Assignments <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>
--	--



Kot Bhalwal, Jammu

Lesson Plan No. 2.1	Course Name: Algorithm Design & Analysis Topic Name: Quick Sort and its analysis	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	Quick Sort and its analysis
Objectives	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> 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)	<ul style="list-style-type: none"> a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> 1. Introduction (5 minutes) <ul style="list-style-type: none"> - Ask questions: <ul style="list-style-type: none"> 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) <ul style="list-style-type: none"> 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.





	<p>i. How to optimize QuickSort so that it takes $O(\log n)$ extra space in worst case?</p> <p>j. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/</p> <p>Quick Sort Algorithm - Divide and Conquer Link : https://youtu.be/IBnyUwCoY4I</p> <p>How Quick Sort Works Performance of Quick Sort with Example Link: https://youtu.be/tWCaFVJMU8</p> <p>k. Summarize the Objectives of learning Quick Sort Algorithm and performing its analysis</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
<p>Closure</p>	<p>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</p> <p>2. Suggested Reading books:</p> <ol style="list-style-type: none"> 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. <p>Home work:</p> <p>3. Activity : Understanding the working of Quick Sort Algorithm and performing its analysis</p> <p>Activity References:</p> <ol style="list-style-type: none"> a. https://www.gatevidyalay.com/tag/quick-sort-in-daa/ b. https://www.codesdope.com/course/algorithms-quicksort/ <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>

Evaluation	<ol style="list-style-type: none">1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.2. Nearpod / Mentimeter short Quiz on Quick Sort Algorithm and its analysis3. MCQ / Sessional Test / Assignments <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>
-------------------	---

Lesson Plan No. 2.2	Course Name: Algorithm Design & Analysis	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	Merge Sort recurrence
Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand how Merge Sort uses a divide and conquer paradigm for sorting. Explaining how algorithm divides the problem into sub problems and solves them individually. Elaborating the process of combining the results of sub problems to get the solution of the original problem Analyse the Time Complexity of Merge Sort(Best, Average and Worst Case). Explaining Merge Sort Applications.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> Ask questions: <ul style="list-style-type: none"> 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. Development (30 minutes) <ol style="list-style-type: none"> 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. Consider an example to understand the approach better. Understand the process of Merging of two lists. Elaborating the Implementation of Merge Sort. Explaining the process how Bottom-Up merge sort approach uses iterative methodology. Analyzing the Time Complexity of Merge Sort recurrence.

	<p>g. Explaining how Merge Sort is useful for sorting linked lists in $O(n \log n)$ time.</p> <p>h. Listing the reasons behind the Slower processing comparative to the other sort algorithms for smaller tasks.</p> <p>i. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/</p> <p>Recursion tree method: intuition Merge Sort Link : https://youtu.be/C4JjXc0htp0</p> <p>Solving the Merge Sort recurrence Link: https://youtu.be/LPfHV3xWwXg</p> <p>j. Summarize the Objectives of learning Merge Sort Algorithm and performing its analysis</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: <ol style="list-style-type: none"> 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. <p>Home work:</p> <ol style="list-style-type: none"> 3. Activity : Understanding the working of Merge Sort Algorithm and performing its analysis <p>Activity References:</p> <ol style="list-style-type: none"> a. https://www.gatevidyalay.com/tag/merge-sort-algorithm-in-daa/ b. https://www.interviewbit.com/tutorial/merge-sort-algorithm/ <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>

Evaluation	<ol style="list-style-type: none">1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.2. Mentimeter short Quiz on Merge Sort Algorithm and its analysis3. MCQ / Sessional Test / Assignments <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>

Lesson Plan No. 2.3	Course Name: Algorithm Design & Analysis Topic Name: Strassen's matrix multiplication	Course No.: PSCSATC-224
----------------------------	--	--------------------------------

Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand the significance of Strassen's Algorithm for matrix multiplication. Explaining how this algorithm is faster than the naive matrix multiplication algorithm. Elaborating the conditions for Strassen's algorithm to work. Understanding the Procedure of Strassen matrix multiplication. Analyse the Time Complexity of Strassen's matrix multiplication. Explaining the Disadvantages of Strassen's matrix multiplication
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> Ask questions: <ul style="list-style-type: none"> 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. Development (30 minutes) <ol style="list-style-type: none"> Discuss the Problem Statement of Naive method during matrix multiplication. Analyzing the Complexity for Naive method of matrix multiplication. Understanding the idea of Strassen's method to reduce the number of recursive calls. Understanding the formula behind the working of algorithm. Calculate the product of A and B (matrix C) with the help of learned formulas. Describing the different conditions for Strassen's algorithm to work.

	<p>g. Understanding how Strassen's matrix multiplication algorithm is asymptotically faster than the naive algorithm.</p> <p>h. Explaining the reasons why generally Strassen's Method is not preferred for practical applications.</p> <p>i. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/</p> <p>Strassen's Matrix Multiplication Link : https://youtu.be/Te08SanNEN8</p> <p>Strassen's Matrix Multiplication - Divide and Conquer - Analysis of Algorithm Link: https://youtu.be/UnpySHwAJsQ</p> <p>j. Summarize the Objectives of learning Strassen's matrix multiplication and performing its analysis</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: <ol style="list-style-type: none"> 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. <p>Home work:</p> <ol style="list-style-type: none"> 3. Activity : Understanding the working of Strassen's matrix multiplication and performing its analysis <p>Activity References:</p> <ol style="list-style-type: none"> a. https://iq.opengenius.org/strassens-matrix-multiplication-algorithm/ b. https://medium.com/swlh/strassens-matrix-multiplication-algorithm-936f42c2b344 <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>

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: Algorithm Design & Analysis Topic Name: Fast multiplication of large integers	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Analysing asymptotic efficiency by ingenious application of the divide-and-conquer technique. Explaining efficient manipulation of large integers. Elaborating the basic idea of the algorithm. Understanding the Procedure Fast multiplication Analyse the Complexity of Fast multiplication and its optimization.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> Ask questions: <ul style="list-style-type: none"> 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. Development (30 minutes) <ol style="list-style-type: none"> Discuss the Problem Statement of Naive method during matrix multiplication. Analyzing the process of addition of long numbers. Understanding the idea of short multiplication: a number times a digit. Understanding the formula behind the working of algorithm. Perform the analysis of long multiplication. Explaining Karatsuba's method for multiplying numbers of any length

	<p>g. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/</p> <p>Large Integer Multiplication - Divide and Conquer - Analysis of Algorithm Link : https://youtu.be/BwHZXuzux44</p> <p>Multiplying Long Integers Using Divide and Conquer Technique Link: https://youtu.be/_YIAcrBwpas</p> <p>h. Summarize the Objectives of learning Multiplication of Long Integers (Faster than Long Multiplication)</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: <ol style="list-style-type: none"> 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. <p>Home work:</p> <ol style="list-style-type: none"> 3. Activity : Understanding the working of Multiplication of Long Integers (Faster than Long Multiplication) <p>Activity References:</p> <ol style="list-style-type: none"> a. https://people.mpiinf.mpg.de/~mehlhorn/ftp/chapter2A-en.pdf b. https://www.brainkart.com/article/Multiplication-of-Large-Integers_8025/ <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<ol style="list-style-type: none"> 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: Algorithm Design & Analysis Topic Name: Binary Search trees	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understanding the nodes arrangement in a Binary Search Tree. Explaining the basic operations associated with a tree. Implementation of Search, Delete and Insert operations in a BST. Understanding the architecture of a basic binary search algorithm. Analyse the Attributes of Binary Search Tree. Explaining the Types of Binary Trees Practice Problems Based on Binary Search Trees
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> Ask questions: <ul style="list-style-type: none"> 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. Development (30 minutes) <ol style="list-style-type: none"> Discuss how we can recursively define a rooted binary tree. Explaining In-Order, PREORDER and POSTORDER Tree Walk traversals along with suitable examples. Understanding the idea of Querying a Binary Search Tree. Understanding the process of Tree Search and Iterative Tree Search. How we can define Successor and predecessor in a BST. Explaining algorithms for Tree Insertion and Deletion. Practice Problems Based on Binary Search Trees Explaining How Good Are Binary Search Trees? Understanding the concept of Balanced Search Trees.

	<p>j. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/</p> <p>Binary Search Tree BST Design & Algorithms Link : https://youtu.be/cvooTtquDIQ</p> <p>Insertion and Traversal in BST Link: https://youtu.be/sXABdGalFNg</p> <p>k. Summarize the Objectives of learning Binary Search trees and doing analysis.</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: <ol style="list-style-type: none"> 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. <p>Home work:</p> <ol style="list-style-type: none"> 3. Activity : Understanding the working of Binary Search trees <p>Activity References:</p> <ol style="list-style-type: none"> a. https://www.gatevidyalay.com/binary-search-trees-data-structures/ b. https://www.studytonight.com/data-structures/binary-search-tree <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<ol style="list-style-type: none"> 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 <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>

Lesson Plan No. 2.6	Course Name: Algorithm Design & Analysis Topic Name: Priority queues	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	Priority queues
Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Introduction to the concept of Priority queues Explaining the basic operations associated with a Priority queue. Understanding how to change priority of an arbitrary element. How to join two Priority queues. Explaining the basic operations associated with a Max Priority queue. Implementation of Priority queues with various techniques.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> Ask questions: <ul style="list-style-type: none"> 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. Development (30 minutes) <ol style="list-style-type: none"> Understanding the process of Assigning Priority Value in a priority queue. Explaining Difference between Priority Queue and Normal Queue. Understanding the Implementation of Priority Queues using different Data Structures. Explaining the process of Inserting an Element into the Priority Queue. Describing the process of Deleting an Element from the Priority Queue.

	<p>f. Understanding concept of Peeking from the Priority Queue (Find max/min).</p> <p>g. Explaining concept of Extract-Max/Min from the Priority Queue.</p> <p>h. Explaining the Priority Queue Applications.</p> <p>i. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/</p> <p>Priority Queue Data Structure Link : https://www.youtube.com/watch?v=NIEwbC6Nt0c</p> <p>Priority Queue - Implementation with Example Link: https://youtu.be/OxhYCLWMdHs</p> <p>j. Summarize the Objectives of learning Binary Search trees and doing analysis.</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: <ol style="list-style-type: none"> 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. <p>Home work:</p> <ol style="list-style-type: none"> 3. Activity : Understanding the working of Priority queues <p>Activity References:</p> <ol style="list-style-type: none"> a. http://www.infocobuild.com/education/audio-video-courses/computerscience/DesignAnalysisOfAlgorithms-CMI/lecture-34.html b. https://algs4.cs.princeton.edu/24pq/ <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<ol style="list-style-type: none"> 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: Algorithm Design & Analysis Topic Name: Heaps, Heap Sort and its analysis	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	Heaps, Heap Sort and its analysis
Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Utilize the Heap to sort the given array. Understanding special heap properties. Describe the concept of Complete and incomplete Binary tree. Create a Heap data structure from unsorted list. Explaining the basic operations associated with Min- Heap and Max Heap. Analyse the Complexity of Heap Sort. Implementing Heap Sort Algorithm.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> Ask questions: <ul style="list-style-type: none"> 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. Development (30 minutes) <ol style="list-style-type: none"> Understanding the conditions required to form a heap. To check whether the binary tree is essentially complete. Understanding the process of implementing heap property. Explaining the process of Inserting an Element into the Priority Queue. Describing the list of important properties of heaps.

	<p>f. Understanding concept of Heap and its array representation.</p> <p>g. How can we construct a heap for a given list of keys?</p> <p>h. Implementation of Heap Sort Algorithm using heapify function.</p> <p>i. Analyzing the Worst Case , Best Case and Average Case Complexity of an algorithm.</p> <p>j. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/</p> <p>Heap Sort Heapify Method Build Max Heap Algorithm Link : https://youtu.be/Q_eia3jC9Ts</p> <p>Heap sort with Example Heapify Method Link: https://youtu.be/nJ6FdAlr_6g</p> <p>k. Summarize the Objectives of learning Heaps, Heap Sort and performing its analysis</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	<p>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</p> <p>2. Suggested Reading books:</p> <ol style="list-style-type: none"> 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. <p>Home work:</p> <p>3. Activity : Practice to implement Heap Sort and learn its analysis</p> <p>Activity References:</p> <ol style="list-style-type: none"> 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/ <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<p>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</p>

-
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

Lesson Plan No. 3.1	Course Name: Algorithm Design & Analysis Topic Name: Data structures for disjoint sets	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	Data structures for disjoint sets
Objectives	At the end of the lesson, the student shall be able to: <ol style="list-style-type: none"> Understand the concept of Union and Find Operation. Keep track of a set of elements partitioned into several disjoint (non-overlapping) subsets. Analyze the worst-case running time of the approach. Illustrate the significance of Union by rank and Path compression.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> What are the basic operations associated with Min- Heap and Max Heap? What are the properties of Complete Binary tree? How we can analyze the Complexity of Heap Sort. Development (30 minutes) <ol style="list-style-type: none"> Understanding the disjoint-set-union data structure along with examples. To manage the operations of union and find. Understanding the process of finding connected components from the graph. Explaining the algorithm for Union by Rank and Path compression. Analyzing the Brute force approach and its complexity with optimized approach. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/ <p>Disjoint Set UNION and FIND</p>

	<p>Link : https://www.youtube.com/watch?v=eTaWFhPXPz4</p> <p>Disjoint Sets using union by rank and path compression Graph Algorithm</p> <p>Link: https://www.youtube.com/watch?v=ID00PMY0-vE</p> <p>g. Summarize the Objectives of learning union by rank and path compression Graph Algorithm.</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: <ol style="list-style-type: none"> 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. <p>Home work:</p> <ol style="list-style-type: none"> 3. Activity : Practice to implement union by rank and path compression <p>Activity References:</p> <ol style="list-style-type: none"> a. https://www.hackerearth.com/practice/data-structures/disjoint-data-structures/basics-of-disjoint-data-structures/tutorial/ <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<ol style="list-style-type: none"> 1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. MCQ / Sessional Test / Assignments <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>

Lesson Plan No. 3.2	Course Name: Algorithm Design & Analysis Topic Name: Concept of Path Compression and Union by Rank	Course No.: PSCSATC-224
----------------------------	---	------------------------------------

Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> To check whether a given graph contains cycle or not To attach smaller depth tree under the root of the deeper tree. Understand the concept of Union and Path Compression. To articulate how size (in place of height) of trees can also be used as <i>rank</i>.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> Explain union() and find() operations for subsets. How to flatten the tree when find() is called. How we can analyze the Time Complexity of Find and Union Operations. Development (30 minutes) <ol style="list-style-type: none"> Using a structure to represent an edge in the graph. Creates a graph with V vertices and E edges To find set of an element i using path compression technique. Understanding the disjoint-set-union data structure along with examples. To manage the operations of union and find. Understanding the process of finding connected components from the graph. Explaining the algorithm for Union by Rank and Path compression. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/ <p>Disjoint Set UNION and FIND Link : https://www.youtube.com/watch?v=eTaWFhPXPz4</p>

	<p>Disjoint Sets using union by rank and path compression Graph Algorithm Link: https://www.youtube.com/watch?v=ID00PMY0-vE</p> <p>i. Summarize the Objectives of learning union by rank and path compression Graph Algorithm.</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	<p>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</p> <p>2. Suggested Reading books:</p> <ul style="list-style-type: none"> 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. <p>Home work:</p> <p>3. Activity : Practice to implement union by rank and path compression</p> <p>Activity References:</p> <ul style="list-style-type: none"> a. https://www.hackerearth.com/practice/data-structures/disjoint-data-structures/basics-of-disjoint-data-structures/tutorial/ b. https://www.geeksforgeeks.org/union-find-algorithm-union-rank-find-optimized-path-compression/ <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<p>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</p> <p>2. MCQ / Sessional Test / Assignments</p> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>

Lesson Plan No. 3.3	Course Name: Algorithm Design & Analysis Topic Name: Prim's and Kruskal's algorithms	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Objectives	At the end of the lesson, the student shall be able to: <ol style="list-style-type: none"> Sort all the edges in non-decreasing order of their weight. Pick the smallest edge. Check if it forms a cycle with the spanning-tree. Build the Minimum Spanning Tree from any vertex in the graph using Prim's Algorithm. Build the Minimum Spanning Tree from the vertex carrying minimum weight in the graph. To Analyse the complexity of Both Algorithms.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> How we can check whether a given graph contains cycle or not What do you understand by Union by Rank? Explain the significance of Path Compression Algorithm. How size of trees can also be used as rank? Development (30 minutes) <ol style="list-style-type: none"> Create a set that keeps track of vertices already included in MST. Assign a key value to all vertices in the input graph. Traversing one node more than one time (prim's algorithm) to get the minimum distance. Traversing one node only once in case of Kruskal's algorithm. Analyzing the differences between Two Algorithms. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/ Prim's Algorithm for Minimum Cost Spanning Tree Prims vs Kruskal

	<p>Link : https://www.youtube.com/watch?v=_KX8GDvRzBc</p> <p>Kruskal Algorithm for Minimum Spanning Tree Link: https://www.youtube.com/watch?v=huQojf2tevI</p> <p>g. Summarize the Objectives of learning union by rank and path compression Graph Algorithm.</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: <ol style="list-style-type: none"> 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. <p>Home work:</p> <ol style="list-style-type: none"> 3. Activity : Practice to implement Prim’s and Kruskal’s algorithms <p>Activity References:</p> <ol style="list-style-type: none"> 1. https://www.gatevidyalay.com/prims-and-kruskal-algorithm-difference/ <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<ol style="list-style-type: none"> 1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. MCQ / Sessional Test / Assignments <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>

Lesson Plan No. 3.4	Course Name: Algorithm Design & Analysis Topic Name: Huffman coding	Course No.: PSCSATC-224
----------------------------	--	--------------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand significance of lossless data compression algorithm. Illustrate the significance of using Fixed Length and Variable Length Codes. Creation and Traversal of Huffman Tree Process of Encoding and Decoding the stream of characters. Analysing the algorithm and finding Time Complexity.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> What do you understand by Spanning Tree? How we can build the Minimum Spanning Tree from any vertex in the graph using Prim's Algorithm. Explain the differences between Prim's and Kruskal's algorithm. How to Analyze the complexity of Both Algorithms. Development (30 minutes) <ol style="list-style-type: none"> Build a Huffman Tree from input characters. Build a Huffman Tree from input characters. Create a leaf node for each unique character and build a min heap of all leaf nodes (Min Heap is used as a priority queue. Extract two nodes with the minimum frequency from the min heap. Create a new internal node with a frequency equal to the sum of the two nodes frequencies. Time Complexity Analysis of Huffman Coding Algorithm. Applications of Huffman Coding. NPTEL Link: https://nptel.ac.in/courses/106/106/106106131/ <p>Huffman Coding Algorithm Greedy Technique Link : https://www.youtube.com/watch?v=uDS8AkTAcIU</p>

	<p>Huffman coding example -Greedy Method Link: https://www.youtube.com/watch?v=saofdNsZiYY</p> <p>i. Summarize the Objectives of learning union by rank and path compression Graph Algorithm.</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: <ol style="list-style-type: none"> 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. <p>Home work:</p> <ol style="list-style-type: none"> 3. Activity : Practice to implement Huffman coding <p>Activity References:</p> <ol style="list-style-type: none"> 1. https://www.gatevidyalay.com/huffman-coding-huffman-encoding/ <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<ol style="list-style-type: none"> 1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. MCQ / Sessional Test / Assignments 3. Nearpod Activity on Huffman coding <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>

Lesson Plan No. 3.5	Course Name: Algorithm Design & Analysis Topic Name: Dijkstra's shortest path algorithm	Course No.: PSCSATC-224
----------------------------	--	--------------------------------

Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand the concept of finding the shortest path in a graph. Learn the steps involved in Dijkstra's algorithm for finding the shortest path. Apply Dijkstra's algorithm to solve real-world problems related to network routing and optimization. Gain proficiency in implementing Dijkstra's algorithm in a programming language of choice.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> - What is the significance of finding the shortest path in a network? - How does Dijkstra's algorithm differ from other graph traversal algorithms? - Can you identify scenarios where Dijkstra's algorithm would be most useful? - What data structures are crucial for the implementation of Dijkstra's algorithm? - How does the algorithm handle graphs with negative edge weights? Development (30 minutes) <ol style="list-style-type: none"> Implement Dijkstra's algorithm in a programming language of your choice. Discuss any challenges faced during the implementation and how they were overcome. Experiment with different strategies to optimize Dijkstra's algorithm. Compare the performance of the original and optimized versions on various graph sizes.

	<p>e. Research and present a case study where Dijkstra's algorithm was crucial for solving a real-world problem.</p> <p>f. NPTEL Link: https://archive.nptel.ac.in/courses/106/106/106106131/</p> <p>YouTubeLink: https://www.youtube.com/watch?v=pVfj6mxhdMw</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: <ol style="list-style-type: none"> 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. <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<ol style="list-style-type: none"> 1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. MCQ / Sessional Test / Assignments 3. Nearpod Activity on Dijkstra's shortest path algorithm <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>

Lesson Plan No. 3.6	Course Name: Algorithm Design & Analysis Topic Name: LZW Coding	Course No.: PSCSATC-224
----------------------------	--	--

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the principles of LZW coding. Implement LZW compression and decompression algorithms. Recognize applications and use cases of LZW coding.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> What is LZW coding, and what problem does it aim to solve in data compression? How does LZW differ from other compression algorithms, such as Huffman coding? Can you explain the basic steps involved in the LZW compression process? What challenges or limitations might be associated with LZW coding? In what real-world scenarios or applications can LZW coding be particularly useful? Development (30 minutes) <ol style="list-style-type: none"> Discuss the LZW compression process step by step. How does it handle repetitive patterns in data? What is the role of the dictionary in LZW coding, and how is it dynamically updated during compression? How does LZW decompression work? Can you outline the key steps involved? Compare LZW coding with other compression algorithms in terms of efficiency and complexity. Explore practical examples where LZW coding has been successfully applied. NPTEL Link: https://archive.nptel.ac.in/courses/106/106/106106131/ <p>YouTubeLink:</p>

	<p>https://www.youtube.com/watch?v=9gkiphD-VIY&t=53s</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	<p>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</p> <p>2. Suggested Reading books:</p> <ul style="list-style-type: none"> 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. <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<p>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</p> <p>2. MCQ / Sessional Test / Assignments</p> <p>3. Nearpod Activity on LZW Coding</p> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



Lesson Plan No. 4.1	Course Name: Algorithm Design & Analysis	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	Dynamic Programming: Basic concepts
Objectives	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> a. Understand the fundamental concepts of Dynamic Programming. b. Explore the advantages of using Dynamic Programming in problem-solving. c. Learn how to identify problems suitable for Dynamic Programming.
Teaching Aids (if any)	<ul style="list-style-type: none"> a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> 1. Introduction (5 minutes) <ul style="list-style-type: none"> Ask questions: <ul style="list-style-type: none"> - How can we break down a problem into subproblems to find an optimal solution?" - Why does identifying and solving overlapping subproblems lead to more efficient solutions? 2. Development (30 minutes) <ul style="list-style-type: none"> a. What characteristics make a problem suitable for a dynamic programming approach? b. How does memoization differ from tabulation in the context of Dynamic Programming? c. In what scenarios would you prefer a bottom-up approach over a top-down approach, and vice versa? d. Can you think of any real-world problems where Dynamic Programming could be a suitable solution? Why? e. NPTEL Link: https://archive.nptel.ac.in/courses/106/106/106106131/ YouTubeLink: https://www.youtube.com/watch?v=0BhhiQGDbEA&t=97s 3. Exercise (10 minutes) – <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books:



	<p>a. Introduction to Algorithms by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition.</p> <p>b. Algorithms by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition.</p> <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<p>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</p> <p>2. MCQ / Sessional Test / Assignments</p> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>

Lesson Plan No. 4.2	Course Name: Algorithm Design & Analysis	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	Matrix-chain multiplication
Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand the concept of matrix chain multiplication. Learn the importance of optimal parenthesization in matrix multiplication. Explore dynamic programming techniques for optimizing matrix chain multiplication. Gain proficiency in implementing matrix chain multiplication algorithms.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <p>Ask questions:</p> <ul style="list-style-type: none"> What is matrix chain multiplication, and why is it relevant in computer science? Can you provide an example where matrix multiplication plays a crucial role? Why is the order of multiplication crucial in matrix multiplication? What challenges can arise if we don't optimize the parenthesizing of matrices? What is dynamic programming, and how does it relate to solving problems like matrix chain multiplication? Can you think of other problems where dynamic programming can be applied? Discuss the time and space complexity of the naive matrix multiplication algorithm. Development (30 minutes) <ol style="list-style-type: none"> Break down the steps involved in solving matrix chain multiplication using dynamic programming. How does the concept of subproblems contribute to the efficiency of the algorithm? Demonstrate the process of finding the optimal parenthesization for a given set of matrices. How can we ensure that the chosen parenthesization is indeed optimal?

	<ol style="list-style-type: none"> e. Walk through the implementation of the matrix chain multiplication algorithm in a programming language of your choice. f. Can you identify potential optimizations or further improvements in the algorithm? g. Explore real-world applications of matrix chain multiplication. h. NPTEL Link: https://archive.nptel.ac.in/courses/106/106/106106131/ <p>YouTubeLink: https://www.youtube.com/watch?v=OnuQ_716lk0&t=105s</p> <ol style="list-style-type: none"> 3. Exercise (10 minutes) – <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: <ol style="list-style-type: none"> 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. <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<ol style="list-style-type: none"> 1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. MCQ / Sessional Test / Assignments 3. Nearpod Activity on Matrix Chain Multiplication <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>

Lesson Plan No. 4.3	Course Name: Algorithm Design & Analysis	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	DP solution for traveling salesman problem
Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand the Traveling Salesman Problem (TSP) and its significance in optimization. Learn the basics of Dynamic Programming (DP) and its application to solve combinatorial optimization problems. Explore the step-by-step approach of developing a DP solution for the Traveling Salesman Problem. Analyze the time and space complexity of the DP solution for TSP. Gain hands-on experience by implementing the DP solution in a programming environment.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> What is the Traveling Salesman Problem, and why is it considered a combinatorial optimization challenge? How does Dynamic Programming differ from other optimization techniques? Can you think of real-world scenarios where finding an optimal route is crucial? Development (30 minutes) <ol style="list-style-type: none"> How does Dynamic Programming address the overlapping subproblems in the context of the TSP? Can you identify the optimal substructure in the Traveling Salesman Problem? How does memoization enhance the efficiency of the DP solution for TSP? NPTEL Link: https://archive.nptel.ac.in/courses/106/106/106106131/ YouTubeLink: https://www.youtube.com/watch?v=3QiSyc7KyC4&t=64s Exercise (10 minutes) – <ul style="list-style-type: none"> Have discussion to summarize the lecture Ask Questions Related to Topics

Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: <ol style="list-style-type: none"> 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. <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<ol style="list-style-type: none"> 1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. MCQ / Sessional Test / Assignments <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>

Lesson Plan No. 4.4	Course Name: Algorithm Design & Analysis	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	DP solution for 0/1 Knapsack problems
Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand the concept of the 0/1 Knapsack Problem. Explore the importance of dynamic programming in solving optimization problems. Learn the step-by-step approach of the dynamic programming solution for the 0/1 Knapsack Problem. Apply the acquired knowledge to solve real-world scenarios using dynamic programming techniques. Analyze the time and space complexity of the dynamic programming solution for the 0/1 Knapsack Problem.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> What is the 0/1 Knapsack Problem, and how does it relate to real-world scenarios? Why is dynamic programming a suitable approach for solving optimization problems? What are the key steps involved in the dynamic programming solution for the 0/1 Knapsack Problem? How does the choice of subproblems contribute to the overall optimization solution? Development (30 minutes) <ol style="list-style-type: none"> Break down the dynamic programming algorithm for the 0/1 Knapsack Problem step by step. Discuss the importance of defining and solving subproblems in this context. Demonstrate the implementation of the dynamic programming solution in a programming language of choice. Discuss how to handle different constraints and variations of the 0/1 Knapsack Problem. Explore ways to optimize the dynamic programming solution in terms of time and space complexity.

	<p>f. Discuss the trade-offs involved in choosing specific optimization strategies.</p> <p>g. NPTEL Link: https://archive.nptel.ac.in/courses/106/106/106106131/</p> <p>YouTubeLink: https://www.youtube.com/watch?v=i8NqAEsZn54</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: <ol style="list-style-type: none"> 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. <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<ol style="list-style-type: none"> 1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. MCQ / Sessional Test / Assignments 3. Nearpod Activity on Knapsack problem <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>

Lesson Plan No. 4.5	Course Name: Algorithm Design & Analysis	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	Concept of least common subsequences and Independent Sets
Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand the concept of Least Common Subsequences (LCS) and its applications. Explore the significance of Independent Sets in various mathematical and computational contexts. Learn the algorithms and techniques for finding LCS and Independent Sets. Apply the acquired knowledge to solve real-world problems related to sequence matching and graph theory.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> What is the purpose of finding the Least Common Subsequence in sequence analysis? How does LCS differ from other sequence-related algorithms? What is the role of Independent Sets in graph theory? Can you provide examples of real-world scenarios where Independent Sets are applicable? How do dynamic programming algorithms contribute to finding the LCS? In what ways do graph algorithms play a role in identifying Independent Sets? How can LCS be applied in bioinformatics or data analysis? Why are Independent Sets important in network design or resource allocation problems? Development (30 minutes) <ol style="list-style-type: none"> Demonstrate the step-by-step process of finding the LCS between two sequences. Discuss the time and space complexity of the algorithm. Explore how to identify Independent Sets in a given graph. Discuss real-world scenarios where finding Independent Sets is crucial. How does dynamic programming contribute to efficient LCS computation?

	<p>f. Can you explain the concept of memoization and its role in LCS algorithms?</p> <p>g. Discuss graph traversal algorithms for identifying Independent Sets.</p> <p>h. Provide examples of graphs where finding Independent Sets is non-trivial.</p> <p>i. NPTEL Link: https://archive.nptel.ac.in/courses/106/106/106106131/</p> <p>YouTubeLink: https://www.youtube.com/watch?v=jHGgXV27qtk&t=237s</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	<p>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</p> <p>2. Suggested Reading books:</p> <ul style="list-style-type: none"> 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. <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>
Evaluation	<p>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</p> <p>2. MCQ / Sessional Test / Assignments</p> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



Lesson Plan No. 4.6	Course Name: Algorithm Design & Analysis	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	Backtracking: General Method
Objectives	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> a. Understand the concept of backtracking in algorithm design. b. Explore the general approach and application of backtracking. c. Learn to implement backtracking to solve combinatorial problems. d. Analyze the time and space complexity of backtracking algorithms.
Teaching Aids (if any)	<ul style="list-style-type: none"> a. PPTs. b. Green board (Chalk and Talk). c. Video Lectures by NPTEL / Youtube d. Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> 1. Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> - What is backtracking, and how does it differ from other algorithmic techniques? - Can you provide examples of real-world problems where backtracking is an effective solution strategy? - How does backtracking handle situations where decisions need to be revised? - What are the key characteristics of problems suitable for a backtracking approach? 2. Development (30 minutes) <ul style="list-style-type: none"> a. Discuss the basic structure of a backtracking algorithm. b. How does backtracking utilize recursion in problem-solving? c. Provide examples of problems that can be solved using the backtracking approach. d. Demonstrate how backtracking can be applied to combinatorial problems such as permutations and combinations. e. Discuss the space requirements and potential optimizations in backtracking. f. Compare the efficiency of backtracking with other problem-solving techniques. g. NPTEL Link: https://archive.nptel.ac.in/courses/106/106/106106131/ YouTubeLink: https://www.youtube.com/watch?v=-47nFq_9_zk&t=38s



	<p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none">- Have discussion to summarize the lecture- Ask Questions Related to Topics
Closure	<p>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</p> <p>2. Suggested Reading books:</p> <ul style="list-style-type: none">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. <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<p>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</p> <p>2. MCQ / Sessional Test / Assignments</p> <p>3. Nearpod Activity on Backtracking algorithm</p> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



Lesson Plan No. 4.7	Course Name: Algorithm Design & Analysis	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	Breadth-First and Depth-First search algorithms
Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand the fundamental concepts of Breadth-First Search (BFS) and Depth-First Search (DFS) algorithms. Differentiate between BFS and DFS and identify scenarios where each algorithm is applicable. Implement BFS and DFS algorithms to traverse graphs and analyze their time complexities. Recognize real-world applications of BFS and DFS in computer science and problem-solving.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> What is graph traversal, and why is it essential in computer science? Can you name some real-world problems where graph traversal is applicable? Can you give an example of a scenario where BFS would be more suitable than DFS? How is recursion utilized in DFS, and what advantages does it offer? In what situations might DFS be a preferred choice over BFS? Can you think of a graph where BFS and DFS would produce different results? Compare the time complexities of BFS and DFS. When might one algorithm be more efficient than the other, and why? Development (30 minutes) <ol style="list-style-type: none"> Develop a recursive algorithm for Depth-First Search. Code a DFS traversal for a given graph. Analyze the time complexity of your BFS and DFS implementations. Discuss the space complexity and memory requirements of both algorithms. Discuss the advantages and disadvantages of each algorithm in specific contexts.



	<p>f. NPTEL Link: https://archive.nptel.ac.in/courses/106/106/106106131/</p> <p>YouTubeLink: https://www.youtube.com/watch?v=N2P7w22tN9c&t=38s</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none">- Have discussion to summarize the lecture- Ask Questions Related to Topics
Closure	<ol style="list-style-type: none">1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.2. Suggested Reading books:<ol style="list-style-type: none">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. <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<ol style="list-style-type: none">1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.2. MCQ / Sessional Test / Assignments <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



Lesson Plan No. 5.1	Course Name: Algorithm Design & Analysis	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	Topological Sort
Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand the concept of topological sorting in directed acyclic graphs (DAGs). Identify scenarios where topological sorting is applicable. Implement topological sorting algorithms. Analyze the time complexity of topological sorting algorithms.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> What is a directed acyclic graph (DAG), and why is it important in the context of topological sorting? How does topological sorting help in identifying the order of tasks or events with dependencies? Can you provide an example scenario where topological sorting would be beneficial? What are the key differences between Kahn's algorithm and depth-first search (DFS) for topological sorting? How does the presence of cycles in a graph affect the possibility of topological sorting? Development (30 minutes) <ol style="list-style-type: none"> Define topological sorting and its significance. Walk through an example to illustrate the concept. Discuss the algorithm's time complexity and its suitability for different scenarios. Compare and contrast DFS with Kahn's algorithm. Provide hands-on exercises for implementing Kahn's algorithm and DFS for topological sorting. Explore real-world applications, such as task scheduling and course prerequisite determination. NPTEL Link: https://archive.nptel.ac.in/courses/106/106/106106131/ YouTubeLink: https://www.youtube.com/watch?v=dis_c84ejhQ&t=35s Exercise (10 minutes) –



	<ul style="list-style-type: none">- Have discussion to summarize the lecture- Ask Questions Related to Topics
Closure	<ol style="list-style-type: none">1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.2. Suggested Reading books:<ol style="list-style-type: none">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. <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<ol style="list-style-type: none">1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.2. MCQ / Sessional Test / Assignments <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>

Lesson Plan No. 5.2	Course Name: Algorithm Design & Analysis	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	Recursive graph algorithms
Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand the concept of recursion in the context of graph algorithms. Explore and implement recursive approaches for common graph operations. Analyze the time and space complexity of recursive graph algorithms. Apply recursive strategies to solve real-world graph-related problems.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> What is recursion, and how does it differ from iteration? Why might recursive algorithms be advantageous in certain graph scenarios? Can you name some common graph operations that could be implemented recursively? How does the recursive approach impact the readability and simplicity of graph algorithms? Development (30 minutes) <ol style="list-style-type: none"> What is the recursive strategy for implementing DFS on a graph? How can we ensure that every vertex is visited exactly once in a recursive DFS? Can you compare the recursive and iterative implementations of DFS in terms of simplicity and efficiency? Discuss the importance of directed acyclic graphs (DAGs) in topological sorting. How can recursion be employed to find connected components in an undirected graph? What are the advantages and limitations of using recursion in maze-solving algorithms? NPTEL Link: https://archive.nptel.ac.in/courses/106/106/106106131/

	<p>YouTubeLink: https://www.youtube.com/watch?v=bJg_sv7PV-g&t=99s</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: <ol style="list-style-type: none"> 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. <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<ol style="list-style-type: none"> 1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. MCQ / Sessional Test / Assignments <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



Lesson Plan No. 5.3	Course Name: Algorithm Design & Analysis	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	String matching: KMP algorithm
Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand the importance of string-matching algorithms in computer science. Comprehend the inefficiencies of naive string-matching algorithms. Introduce the Knuth-Morris-Pratt (KMP) algorithm as an efficient string-matching solution. Learn the key components of the KMP algorithm and how they contribute to its efficiency. Implement the KMP algorithm in practical coding scenarios.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> - Why is string matching important in computer science and real-world applications? - What are the limitations of naive string-matching algorithms, and why do we need more efficient solutions? - How does the KMP algorithm improve the efficiency of string matching? - What are the main components of the KMP algorithm, and how do they work together? - Can you outline the steps involved in implementing the KMP algorithm in a programming language of your choice? Development (30 minutes) <ol style="list-style-type: none"> Discuss the concept of string matching and its applications in various fields. Explain the inefficiencies of naive string-matching algorithms, highlighting their time complexity. Introduce the Knuth-Morris-Pratt (KMP) algorithm and its historical significance. Break down the key components of the KMP algorithm, including the pre-processing step and the actual search process. Guide students through the implementation of the KMP algorithm in a step-by-step manner. Encourage discussions on real-world scenarios where the KMP algorithm could be advantageous.



	<p>g. NPTEL Link: https://archive.nptel.ac.in/courses/106/106/106106131/</p> <p>YouTubeLink: https://www.youtube.com/watch?v=GTJr8OvyEVQ&t=157s</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none">- Have discussion to summarize the lecture- Ask Questions Related to Topics
Closure	<ol style="list-style-type: none">1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.2. Suggested Reading books:<ol style="list-style-type: none">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. <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<ol style="list-style-type: none">1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.2. MCQ / Sessional Test / Assignments <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



Lesson Plan No. 5.4	Course Name: Algorithm Design & Analysis	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	Rabin-Karp algorithm
Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand the basics of string searching algorithms. Explore the Rabin-Karp algorithm and its significance in pattern matching. Learn the key concepts of hash functions and their role in the Rabin-Karp algorithm. Implement the Rabin-Karp algorithm in a practical example.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> Why is string matching important in computer science and applications? Can you think of real-world scenarios where efficient string matching is crucial? How does it utilize hashing to expedite string matching? What are the primary components of the Rabin-Karp algorithm? How does the rolling hash function work in the context of this algorithm? In what situations does the Rabin-Karp algorithm outperform other string matching algorithms? Development (30 minutes) <ol style="list-style-type: none"> Demonstrate the step-by-step process of implementing the Rabin-Karp algorithm for a given string matching problem. How can you handle collisions in the hash function? How does the choice of the hash function impact the algorithm's performance? Explore possible optimizations or variations of the Rabin-Karp algorithm. Can you modify the algorithm to accommodate different types of patterns or text sources? NPTEL Link: https://archive.nptel.ac.in/courses/106/106/106106131/



	<p>YouTubeLink: https://www.youtube.com/watch?v=H4VrKHVG5ql&t=135s</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none">- Have discussion to summarize the lecture- Ask Questions Related to Topics
Closure	<ol style="list-style-type: none">1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.2. Suggested Reading books:<ol style="list-style-type: none">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. <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<ol style="list-style-type: none">1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.2. MCQ / Sessional Test / Assignments3. Nearpod Activity on Rabin-Carp Algorithm <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>

Lesson Plan No. 5.5	Course Name: Algorithm Design & Analysis	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	Polynomial Time Algorithms
Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand the concept of polynomial time algorithms in the context of computational complexity theory. Recognize the significance of polynomial time algorithms in solving computational problems efficiently. Explore common examples of polynomial time algorithms and their applications. Analyze the difference between polynomial time and non-polynomial time algorithms. Evaluate the efficiency and scalability of algorithms through the lens of polynomial time complexity.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> How does polynomial time differ from non-polynomial time, and why is this classification important? Can you give an example of a problem that is known to have a polynomial time solution? In what scenarios would we prefer using polynomial time algorithms over non-polynomial time algorithms? Development (30 minutes) <ol style="list-style-type: none"> Compare and contrast polynomial time algorithms with non-polynomial time algorithms. Explore examples of well-known polynomial time algorithms, such as sorting algorithms or graph algorithms. Analyze the implications of P vs NP problem in the context of polynomial time algorithms. Discuss the significance of polynomial time algorithms in practical problem-solving scenarios. NPTEL Link: https://archive.nptel.ac.in/courses/106/106/106106131/ <p>YouTubeLink: https://www.youtube.com/watch?v=ZSyZozw6JMQ&t=347s</p>

	<p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading books: <ol style="list-style-type: none"> 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. <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<ol style="list-style-type: none"> 1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. MCQ / Sessional Test / Assignments 3. Nearpod Activity <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>

Lesson Plan No. 5.6	Course Name: Algorithm Design & Analysis	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	Basics of number Theory Algorithms
Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand fundamental concepts of number theory. Explore basic number theory algorithms and their applications. Develop problem-solving skills related to number theory challenges. Gain proficiency in implementing number theory algorithms in programming.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> What is number theory, and why is it important in computer science? Can you provide real-world examples where number theory is applied? What is a prime number, and how can we determine if a number is prime? How do factors and multiples relate to number theory? What is the Euclidean algorithm, and how does it find the greatest common divisor (GCD)? How can we use the Sieve of Eratosthenes to generate prime numbers efficiently? In what ways are number theory algorithms used in cryptography? Can you think of examples where modular arithmetic is applied in computing? Development (30 minutes) <ol style="list-style-type: none"> Discuss the historical significance of number theory in the development of algorithms. Explore how number theory forms the foundation for various cryptographic techniques. Break down the steps involved in determining whether a number is prime or composite. Investigate the relationship between prime factorization and unique factorization in number theory.

	<p>e. Implement the Euclidean algorithm in a programming language of your choice.</p> <p>f. Demonstrate the step-by-step process of the Sieve of Eratosthenes for finding prime numbers.</p> <p>g. NPTEL Link: https://archive.nptel.ac.in/courses/106/106/106106131/</p> <p>YouTubeLink: https://www.youtube.com/watch?v=69jsFIMINpl&t=35s</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none"> - Have discussion to summarize the lecture - Ask Questions Related to Topics
Closure	<p>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</p> <p>2. Suggested Reading books:</p> <ul style="list-style-type: none"> 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. <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<p>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</p> <p>2. MCQ / Sessional Test / Assignments</p> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



Lesson Plan No. 5.7	Course Name: Algorithm Design & Analysis	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	GCD and extended Euclidean algorithm
Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand the concept of Greatest Common Divisor (GCD) and its significance in number theory. Learn the basic algorithm for finding the GCD of two numbers. Explore the Extended Euclidean Algorithm and its application in finding modular inverses. Apply the Extended Euclidean Algorithm to solve linear Diophantine equations. Gain practical insights into the applications of GCD and Extended Euclidean Algorithm in computer science and cryptography.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> What is the significance of finding the Greatest Common Divisor (GCD) in number theory? Can you provide real-world examples where GCD plays a crucial role? How would you compute the GCD of two numbers using prime factorization? Can you think of more efficient ways to find the GCD without prime factorization? Why might we need an extended version of the Euclidean Algorithm? How can the Extended Euclidean Algorithm be used to find modular inverses? In what areas of computer science or cryptography might we encounter problems that involve the Extended Euclidean Algorithm? How does the Extended Euclidean Algorithm help in solving linear Diophantine equations? Development (30 minutes) <ol style="list-style-type: none"> Demonstrate the step-by-step process of finding the GCD of two numbers using the Euclidean Algorithm. How can you use the GCD to simplify fractions?



	<p>c. Illustrate the Extended Euclidean Algorithm with a numerical example.</p> <p>d. Explain how the Extended Euclidean Algorithm can be applied to find modular inverses.</p> <p>e. Discuss a specific example where GCD is used in computer science, programming, or cryptography.</p> <p>NPTEL Link: https://archive.nptel.ac.in/courses/106/106/106106131/</p> <p>YouTubeLink: https://www.youtube.com/watch?v=yHwneN6zJmU&t=10s</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none">- Have discussion to summarize the lecture- Ask Questions Related to Topics
Closure	<p>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</p> <p>2. Suggested Reading books:</p> <ul style="list-style-type: none">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. <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<p>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</p> <p>2. MCQ / Sessional Test / Assignments</p> <p>3. Nearpod Activity</p> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



Lesson Plan No. 5.8	Course Name: Algorithm Design & Analysis	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	Primality Testing
Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Students will understand the concept of primality testing and be able to identify prime numbers. Familiarize students with various primality testing algorithms. Enable students to implement simple primality tests in programming languages. Develop critical thinking skills in analyzing the efficiency of different primality testing methods.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> Definition and characteristics of prime numbers. Why are prime numbers important in cryptography? What is the Sieve of Eratosthenes? Can you think of real-world applications where primality testing is crucial? Discuss applications in computer science, cryptography, and number theory. How do primality testing algorithms differ in terms of efficiency? Development (30 minutes) <ol style="list-style-type: none"> Break down the steps involved and discuss the time complexity. Discuss the underlying principles and conditions for Fermat's test. Address issues like false positives and false negatives in testing methods. Introduce the concept of probabilistic primality testing. Discuss algorithms like Miller-Rabin and their trade-offs between accuracy and efficiency. Encourage students to write a simple program to test primality and discuss optimization techniques. NPTEL Link:



	<p>https://archive.nptel.ac.in/courses/106/106/106106131/</p> <p>YouTubeLink: https://www.youtube.com/watch?v=sDrXeCs3ghQ&t=209s</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none">- Have discussion to summarize the lecture- Ask Questions Related to Topics
Closure	<ol style="list-style-type: none">1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.2. Suggested Reading books:<ol style="list-style-type: none">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. <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<ol style="list-style-type: none">1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.2. MCQ / Sessional Test / Assignments3. Nearpod Activity on LZW Coding <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



Lesson Plan No. 5.9	Course Name: Algorithm Design & Analysis	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	Basic Concepts and Non-Deterministic Algorithms
Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand the fundamental concepts of algorithms and their significance in problem-solving. Differentiate between deterministic and non-deterministic algorithms. Grasp the basic characteristics and applications of non-deterministic algorithms. Evaluate scenarios where non-deterministic algorithms may be more suitable.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> What is the role of algorithms in computer science and problem-solving? How would you define a deterministic algorithm? Can you provide an example of a problem where a deterministic algorithm might not be the most efficient solution? What distinguishes non-deterministic algorithms from deterministic ones? In what real-world scenarios can non-deterministic algorithms be beneficial? What are some basic characteristics of non-deterministic algorithms? How do they handle randomness or uncertainty in problem-solving? Development (30 minutes) <ol style="list-style-type: none"> Explore the concept of non-deterministic algorithms and how they differ from deterministic counterparts. Investigate practical applications of non-deterministic algorithms in fields such as optimization, cryptography, or artificial intelligence. Discuss how non-deterministic algorithms leverage probability and randomness to arrive at solutions.



	<p>d. Introduce the concept of NP-completeness and its relevance to non-deterministic algorithms.</p> <p>e. Examine situations where non-deterministic algorithms might offer efficiency advantages over deterministic alternatives.</p> <p>f. NPTEL Link: https://archive.nptel.ac.in/courses/106/106/106106131/</p> <p>YouTubeLink: https://www.youtube.com/watch?v=ZNe1ziMExGg&t=273s</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none">- Have discussion to summarize the lecture- Ask Questions Related to Topics
Closure	<p>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</p> <p>2. Suggested Reading books:</p> <ul style="list-style-type: none">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. <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<p>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</p> <p>2. MCQ / Sessional Test / Assignments</p> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



Lesson Plan No. 5.10	Course Name: Algorithm Design & Analysis	Course No.: PSCSATC-224
-----------------------------	---	--------------------------------

Topics	Polynomial Time and algorithms
Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Define and understand the concept of polynomial time in the context of algorithms. Differentiate between polynomial and non-polynomial time complexity. Identify algorithms with polynomial time complexity. Analyze the efficiency of algorithms using Big-O notation.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> Definition and implications. Examples of polynomial time algorithms (e.g., bubble sort, merge sort). Brief introduction to non-polynomial time complexity (e.g., exponential time complexity). Understanding the implications of non-polynomial time. Explanation of Big-O notation as a tool for describing algorithmic complexity. Practical examples illustrating Big-O notation. Development (30 minutes) <ol style="list-style-type: none"> Analyze the time complexity of a specific algorithm (e.g., binary search) and determine if it is polynomial. Compare the time complexity of two sorting algorithms, one with polynomial time complexity and the other with non-polynomial time complexity. Discuss a real-world problem and propose an algorithm to solve it, considering the importance of polynomial time. NPTEL Link: https://archive.nptel.ac.in/courses/106/106/106106131/ <p>YouTubeLink: https://www.youtube.com/watch?v=VVStIz30V8&t=36s</p> Exercise (10 minutes) – <ul style="list-style-type: none"> Have discussion to summarize the lecture Ask Questions Related to Topics



Closure	<ol style="list-style-type: none">1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.2. Suggested Reading books:<ol style="list-style-type: none">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. <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<ol style="list-style-type: none">1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.2. MCQ / Sessional Test / Assignments3. Nearpod Activity <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



Lesson Plan No.5.11	Course Name: Algorithm Design & Analysis	Course No.: PSCSATC-224
----------------------------	---	--------------------------------

Topics	NP-hard & NP –complete classes
Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand the concepts of NP-Hard and NP-Complete problems. Differentiate between problems that belong to P, NP, NP-Hard, and NP-Complete classes. Analyze the implications of NP-Hard and NP-Complete problems on computational complexity. Explore strategies for dealing with NP-Complete problems.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> What are complexity classes, and why are they essential in computational theory? Can you provide examples of problems that are easy to solve (in P) and those that are hard to solve (beyond P)? What is the definition of an NP-Hard problem? How does an NP-Hard problem differ from an NP-Complete problem? What does it mean for a problem to be NP-Complete? Can you explain how Cook's theorem is related to NP-Completeness? How does the verification of a solution relate to NP problems? Why is it easier to verify a solution than to find one for NP problems? Development (30 minutes) <ol style="list-style-type: none"> Discuss examples of problems that are known to be NP-Hard. How can reductions be used to show that a problem is NP-Hard? Analyze real-world problems that fall into the NP-Complete category. Can you identify situations where dealing with NP-Complete problems is unavoidable? Explain the significance of Cook's theorem in establishing NP-Completeness. How does Cook's theorem impact the way we understand the complexity of computational problems?



	<p>g. Explore heuristic methods and approximation algorithms for solving NP-Complete problems.</p> <p>h. NPTEL Link: https://archive.nptel.ac.in/courses/106/106/106106131/</p> <p>YouTubeLink: https://www.youtube.com/watch?v=sJT1Zzubz5A&t=11s</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none">- Have discussion to summarize the lecture- Ask Questions Related to Topics
Closure	<p>1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.</p> <p>2. Suggested Reading books:</p> <ul style="list-style-type: none">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. <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<p>1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.</p> <p>2. MCQ / Sessional Test / Assignments</p> <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>

Lesson Plan No. 5.12	Course Name: Algorithm Design & Analysis	Course No.: PSCSATC-224
-----------------------------	---	--------------------------------

Topics	Cook's Theorem
Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand the fundamental concepts of Cook's Theorem in computational complexity theory. Explore the significance of Cook's Theorem in the context of the theory of computation. Gain insights into the relationship between the complexity classes P and NP. Apply the key ideas from Cook's Theorem to analyze and categorize computational problems.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> What is the primary focus of Cook's Theorem? How does Cook's Theorem contribute to our understanding of computational complexity? Can you explain the concept of polynomial-time reducibility? Development (30 minutes) <ol style="list-style-type: none"> How does Cook's Theorem relate to the P versus NP problem? Discuss any real-world implications or applications of Cook's Theorem. NPTEL Link: https://archive.nptel.ac.in/courses/106/106/106106131/ YouTubeLink: https://www.youtube.com/watch?v=JYBoaoPkaQM&t=63s Exercise (10 minutes) – <ul style="list-style-type: none"> Have discussion to summarize the lecture Ask Questions Related to Topics
Closure	<ol style="list-style-type: none"> Summarize the Lesson Learning Outcomes and get affirmation from students on these. Suggested Reading books: <ol style="list-style-type: none"> Introduction to Algorithms by T.Cormen, C. Lieserson, R.Rivest, C.Steina. Prentice-Hall/India, 3rd Edition.

	<p>b. Algorithms by S. Dasgupta, C. Papadimitriou, Umesh Vazirani. McGraw Hill Education, 1st Edition.</p> <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
<p>Evaluation</p>	<ol style="list-style-type: none"> 1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. MCQ / Sessional Test / Assignments 3. Nearpod Activity <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>

Lesson Plan No. 5.13	Course Name: Algorithm Design & Analysis	Course No.: PSCSATC-224
-----------------------------	---	--------------------------------

Topics	Introduction to Approximation Algorithms
Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand the concept of approximation algorithms and their significance in solving optimization problems. Explore the trade-off between accuracy and efficiency in algorithm design. Recognize real-world applications where approximation algorithms play a crucial role. Analyze the performance guarantees and bounds associated with approximation algorithms.
Teaching Aids (if any)	<ol style="list-style-type: none"> PPTs. Green board (Chalk and Talk). Video Lectures by NPTEL / Youtube Use of Nearpod / Mentimeter / Kahoot tools.
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Ask questions: <ul style="list-style-type: none"> - What is the primary goal of optimization problems in computer science? - Why are exact solutions often impractical for NP-hard problems? - Can you provide an example of a real-world problem that can be modeled as an optimization problem? - How do approximation algorithms strike a balance between optimality and efficiency? - What are the challenges associated with designing and analyzing approximation algorithms? Development (30 minutes) <ol style="list-style-type: none"> How do we measure the performance of an approximation algorithm? Discuss the concepts of approximation ratio and approximation factor. How do greedy algorithms contribute to approximation algorithms? Provide examples where greedy strategies can be applied. Explain the concept of NP-hardness. Discuss why finding exact solutions for NP-hard problems is often infeasible.



	<ul style="list-style-type: none">g. Explore specific approximation algorithms for well-known problems (e.g., Traveling Salesman Problem, Knapsack Problem).h. Discuss their applications and limitations.i. NPTEL Link: https://archive.nptel.ac.in/courses/106/106/106106131/ <p>YouTubeLink: https://www.youtube.com/watch?v=u5rqrQqGk3E&t=16s</p> <p>3. Exercise (10 minutes) –</p> <ul style="list-style-type: none">- Have discussion to summarize the lecture- Ask Questions Related to Topics
Closure	<ul style="list-style-type: none">1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.2. Suggested Reading books:<ul style="list-style-type: none">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. <p>Spend 5 minutes to wrap up and consolidate the leanings.</p>
Evaluation	<ul style="list-style-type: none">1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.2. MCQ / Sessional Test / Assignments <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>