c. Rotation
d. Shrinking
e. Zooming

0. To understand various image noise models and to write programs for:

a. Image restoration
b. Remove Salt and Pepper Noise
c. Minimize Gaussian noise
d. Median filter and Weiner filter

0. Write and execute programs to remove noise from images using spatial filtering.

a. Understand 1-D and 2-D convolution process
b. Use 3x3 Mask for low pass filter and high pass filter

0. Write and execute programs for image frequency domain filtering.

a. Apply FFT on given image
b. Perform low pass and high pass filtering in frequency domain
c. Apply IFFT to reconstruct image

0. Write and execute a program for edge detection using different edge detection mask.

0. Write and execute a program for image morphological operations erosion and dilation

DISCIPLINE SPECIFIC ELECTIVE COURSE: Advanced Algorithms

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course (if any)
		Lecture	Tutorial	Practical/ Practice		
DSE7b: Advanced Algorithms	4	3	1	0	Pass in Class XII	Design and Analysis of Algorithms

Course Objective

This course is designed to provide exposure to more sophisticated algorithms for some tractable problems, some advanced topics in algorithms such as NP Completeness and how to handle NP hard problems in practice.

Learning Outcomes

On successful completion of the course, students will be able to:

1. Understand and develop more sophisticated algorithms using some of the nown design techniques.
2. Identify NP hard problems.
3. Use polynomial time reductions to prove NP hardness of problems.
4. Design approximation algorithms for NP hard problems and find their approximation ratio.

Syllabus

Unit 1 More applications of Divide and Conquer, Greedy and Dynamic Programming approaches: Counting Inversions, Closest pair of points, Integer Multiplication, Huffman Code, Segmented Least Squares etc.

Unit 2 Network Flows: Ford Fulkerson algorithm for max flow problem.

Unit 2 Backtracking: Constructing All Subsets, Constructing All Permutations, Constructing all paths in a graph.

Unit 3 Polynomial time reductions via gadgets: SAT and 3-SAT problems; Reducing 3-SAT to Independent set, Clique and Vertex cover.

Unit 4 Proving NP completeness: Circuit satisfiability, 3-SAT, Sequencing Problems, Graph coloring, Subset sum.

Unit 5 Introduction to Approximation Algorithms: Definition, Concept of approximation factor, Bounding the optimal solution, concept of tight example.

Unit 6 Combinatorial Approximation Algorithms: Set cover, Minimizing makespan, k-center.

Unit 7 LP based Approximation Algorithms: Approximation algorithms for Vertex cover/Set cover via LP rounding.

References

1. Kleinberg, J., Tardos, E. *Algorithm Design,* 1st edition, Pearson, 2013.
2. Vazirani, V. V. *Approximation Algorithms,* 1st edition, Springer, 2001.

Additional References

(i) Cormen, T.H., Leiserson, C.E., Rivest, R. L., Stein C. *Introduction to Algorithms*, 4th edition, Prentice Hall of India, 2022.

(ii) Williamson, D. P., Shmoys, D. B. *The Design of Approximation Algorithms*, 1st edition, Cambridge University Press, 2011.

Tutorials

Tutorials based on Theory