Algorithm Analysis & Design - 2
Session: January'2025 - May'2025

1. Course Number and Name:

CSE 2632, Algorithm Design 2

2. Credits and Course Format:

Credits = 4
3 Classes/week, 1hr/Class,
1 Lab/Week, 2hr/Lab

3. Target Students:

Programme: B.Tech. (4^{th} Semester)
Branch: CSE, CS&IT, CSE(AI&ML), CSE(IoT), CSE(Cyber Security), CSE(Data Science)

4. Instructor's Names:

1. Satya Ranjan Das, CSE, ITER
2. Paresh Baidya, CSE, ITER
3. Binayak Panda, CSE, ITER
4. Ranga Ballav Pradhan, ITER
5. Rajesh Purkait, CSE, ITER
6. Pubali Chatterjee, CSE, ITER
7. Irshed Hussain, CSIT, ITER
8. Sudeepa Das, CSIT, ITER
9. Swagat Kumar Jena, CSE, ITER
10. Brajesh Kumar Umrao, CSE, ITER
11. Kartik Sahoo, CSE, ITER
12. Shruti Bajpai, CSE, ITER
13. Soumyashree Panda, CSE, ITER
14. Sandhyarani Sahoo, CSE, ITER
15. Samir Kundu, CSE, ITER
16. Kundan Chandra Patra, CSE, ITER
17. Debapriya Panda, CSE, ITER
18. Amit Kumar Singh, CSE, ITER
19. Arup Kumar Dutta, CSE. ITER
20. Sabuzima Nayak, CSE. ITER
21. Manas Kumar Swain, CSE, ITER
22. Rakesh Sen, CSE, ITER
23. Geet Sahoo, CSE, ITER

5. Text Book and References:

Text book

1. (T1) Algorithm Design by Eva Tardos, J. Kleinberg, Pearson Publication
2. (T2) Problem Soving in Data Structures and Algorithms using Java by Hemant Jain

Reference book

1. (P1) Introduction to Algorithms by T.M. Coremen, C.E. Leiserson, R.L. Rivest, C. Stein, PHI Publication (CLRS)
2. (P2) The Algorithm Design Manual by Steven S. Skiena, Springer Publication

6. Specific Course Information:

a. Course Description:

The course topics will include various algorithmic design patterns like dynamic programming, backtracking, randomized algorithms. Course will also cover major algorithms and data structures for searching and sorting, graphs, and some optimization techniques.

b. Prerequisites and/or Co-requisites:

Prerequisite: CSE 2631(AAD1), CSE 1002(DM)
Co-requisite: CSE 2041(CSW2)

7. Course Learning Outcomes:

By the end of course through lectures, readings, homeworks, lab and theory assignments and exams, students will be able to:

CO1: understand the network flow problem and apply it to real-world problems.
CO2: - distinguish between computationally tractable and intractable problems.
- define and relate class-P, class-NP and class NP-complete, PSPACE, PSPACE-complete
- given a problem in NP, define an appropriate certificate and the verification algorithm.
CO3: understand approximation algorithms and apply this concept to solve problems.
CO4: understand local search techniques and apply this concept to solve problems.
CO5: understand randomization and apply this concept to solve problems.
CO6: identify and apply an appropriate algorithmic approach to solve a problem and explain the challenges to solve it.

8. Brief List of Topics to Be Covered: (L: Lecture, P: Practical/Problem Solving Session)

- Network Flow Problem and Its Applications
- NP and Computational Intractability
- PSPACE: A class Problems Beyond NP
- Extending the Limits of Intractability
- Approximation Algorithms
- Local Search
- Randomized Algorithms

➢

Contact Hour	Topics To Be Covered	Remarks(if any)
Week # 1:		
L 01	Introduction to the course/subject: Lesson plan; Course Goal; Teaching methodology; Evaluation strategy etc.	
L 02	Review on Time and Space Complexity, Asymptotic Notations, Recurrences, Sorting and Searching, (Through discussion and some practice questions.)	
L 03	Review on Greedy, DAC and DP Algorithms and Review on Graph representation, graph traversal, spanning tree and shortest path (Through discussion and some practice questions.)	
P 01	*Java implementation of Graph related algorithms (BFS, DFS, MST, Shortest Path)* [Chapter-12: Graphs- Example-12.1, 12.2, 12.3, 12.4, 12.5, 12.7, 12.8, 12.9, 12.13, 12.14, 12.18, 12.21, 12.24]	To be referred from T2 (Chapter-12)
Week # 2:		
L 04	Weighted graph algorithms: Network flows and Bipartite matching. [Section- 7.1 (Page: 337-345) including theorems and lemma – 7.1, 7.3, 7.5 statements only with example, no proof required]	
L 05	Weighted graph algorithms: Network flows and Bipartite matching [Section- 7.2 (Page: 346-352) including theorems and lemma – 7.6, 7.7, 7.8 statements only with example, no proof required]	To be referred from T1(Chapter-7.1, 7.2)
L 06	Weighted graph algorithms: Network flows and Bipartite matching [Section- 7.2 (Page: 346-352) including theorem–7.9 proof required and discuss How to choose good augmenting path]	To be referred from T1(Chapter-7.3)
P 02	*Java implementation of Graph related algorithms (BFS, DFS, MST, Shortest Path)* [Chapter-12: Graphs- Example-12.1, 12.2, 12.3, 12.4, 12.5, 12.7, 12.8, 12.9, 12.13, 12.14, 12.18, 12.21, 12.24]	To be referred from T2 (Chapter-12)
Week # 3:		
L 07	Weighted graph algorithms: Network flows and Bipartite matching [Section- 7.5 (Page: 367-370) including lemmas –7.34, 7.35, 7.36, 7.37 statements only with example, no proof required]	To be referred from T1(Chapter-7.5)
L 08	NP and Computational Intractability: Problem Reductions (all subsections)	To be referred from P2(Chapter-9.2)
L 09	NP and Computational Intractability: Problem Reductions (all subsections)	To be referred from P2(Chapter-9.3)
P 03	*Java implementation of String-based algorithms (Brute-force algorithm for searching, Rabin-Karp string matching algorithm, Symbol table/Dictionary,Hash table)*	To be referred from T2(Chapter-13)
Week # 4:		
L 10	NP and Computational Intractability: Problem Reductions (all subsections)	To be referred from

		T1(Chapter-8.1)
L 11	NP and Computational Intractability: Satisfiability (Formula-SAT, CKT-SAT etc.) (including theorems and lemma – 8.8, 8.9, 8.13, 8.15)	To be referred from T1(Chapter-8.2)
L 12	NP and Computational Intractability: P, NP, NPC and NPH (including theorems and lemma – 8.10, 8.11, 8.12, 8.16)	To be referred from T1(Chapter-8.3,8.4)
P 04	*Java implementation of String-based algorithms (Brute-force algorithm for searching, Rabin-Karp string matching algorithm, Symbol table/Dictionary,Hash table)*	To be referred from T2(Chapter-13)
Week # 5:		
L 13	NP and Computational Intractability: P, NP, NPC and NPH	To be referred from T1(Chapter-8)
L 14	PSPACE: A class Problems Beyond NP (including all theorems and lemma – 9.1 to 9.3)	To be referred from T1(Chapter-9.1)
L 15	PSPACE: A class Problems Beyond NP	To be referred from T1(Chapter-9.2)
P 05	*Java implementation of String-based algorithms (Brute-force algorithm for searching, Rabin-Karp string matching algorithm, Symbol table/Dictionary,Hash table)*	To be referred from T2(Chapter-13)
Week # 6:		
L 16	PSPACE: A class Problems Beyond NP	To be referred from T1(Chapter-9.2)
L 17	Extending the Limits of Intractability (including theorems and lemma – 10.1 to 10.4 – statements only with example, no proof required)	To be referred from T1(Chapter-10.1)
L 18	Extending the Limits of Intractability (including theorems and lemma – 10.1 to 10.4 – statements only with example, no proof required)	To be referred from T1(Chapter-10.1)
P 06	*Java implementation of Greedy Algorithm (Interval Scheduling, Fractional Knapsack, Huffman Coding)*	To be referred from T2(Chapter-16)
Week # 7:		
L 19	Approximation Algorithms (including all theorems and lemma -- 11.1 to 11.5)	To be referred from T1(Chapter-11.1)
L 20	Approximation Algorithms (including all theorems and lemma -- 11.6 to 11.8)	To be referred from

		T1(Chapter-11.2)
L 21	Approximation Algorithms (including only the statements of all theorems and lemma -- 11.9 to 11.10—no proofs required)	To be referred from T1(Chapter-11.3)
P 07	*Java implementation of Greedy Algorithm (Interval Scheduling, Fractional Knapsack, Huffman Coding)*	To be referred from T2(Chapter-16)
Week # 8:		
L 22	Approximation Algorithms (including all theorems and lemma -- 11.12 to 11.15)	To be referred from T1(Chapter-11.4)
L 23	Approximation Algorithms	Overall discussion from T1(Chapter-11)
L 24	Local Search:	To be referred from T1(Chapter-12.1)
P 08	*Java implementation of Divide-and-Conquer based algorithms (Merge sort, Counting Inversions, Quick sort)*	To be referred from T2(Chapter-17)
Week # 9:		
L 25	Local Search (The landscape of an optimization problem – statement 12.1 with example of vertex cover problem)	To be referred from T1(Chapter-12.1)
L 26	Local Search (Application of local serach to Hopfield Neural Networks – algorithm with example – no proof required)	To be referred from T1(Chapter-12.3)
L 27	Local Search (Application of local serach to Hopfield Neural Networks – algorithm with example – no proof required)	To be referred from T1(Chapter-12.3)
P 09	*Java implementation of Divide-and-Conquer based algorithms (Merge sort, Counting Inversions, Quick sort)*	To be referred from T2(Chapter-17)
Week # 10:		
L 28	Randomized Algorithm: Contention Resolution	To be referred from T1(Chapter-13.1)
L 29	Randomized Algorithm: Median Finding and Quick Sort	To be referred from T1(Chapter-13.5)
L 30	Randomized Algorithm: Hashing	To be referred from

		T1(Chapter-13.6)
P 10	*Java implementation of Dynamic Programming based algorithms (Weighted Intrval Scheduling, Longest Common Subsequence, Coin Exchange, Matrix Chain Multiplication)*	To be referred from T2(Chapter-18)
Week # 11:		
L 31	How to Design Algorithms; A catalog of Algorithmic Problems	To be referred from P2(Chapter-10,11)
L 32	How to Design Algorithms; A catalog of Algorithmic Problems	To be referred from P2(Chapter-10,11)
L 33	How to Design Algorithms; A catalog of Algorithmic Problems	To be referred from P2(Chapter-10,11)
P 11	*Java implementation of Dynamic Programming based algorithms (Weighted Intrval Scheduling, Longest Common Subsequence, Coin Exchange, Matrix Chain Multiplication)*	To be referred from T2(Chapter-18)
Week # 12:		
L 34	Revision Class	Revision Class
L 35	Revision Class	Revision Class
L 36	Revision Class	Revision Class
P 12	Review of lab assignments	

9. Evaluation scheme (under Grading Pattern-1):

Assignments+Term-Projects :	10% + 10%
Attendance:	5%
Mid-semester:	15%
End-semester(Lab. Test):	15%
End-semester(Theory):	45%

10. Teaching Pedagogy/ICT Tools:

We will adopt the basic rules and norms (take notes; participate by asking and answering questions; conduct regular quizzes). The basis of teaching remains the same - content, pedagogy,

and assessment. Careful planning on these three factors remains central to successful teaching - both residential and online. The forms of teaching interaction may change: even as the online medium precludes certain forms of in-person contact, it creates opportunities for new ways to interact.