Algorithm Design-1
Session: March'2021 - May'2021

1. Course Number and Name:

CSE 3131, Algorithm Design1

2. Credits and Course Format:

4 Credits;
3 Classes/Week, 1 hr/Class;
1 Problem Solving Session/Week, 2 hrs/Problem Solving Session

3. Target Students:

Programme: B.Tech. (4^{th} Semester)
Branch: CSE, CS&IT

4. Instructor's Names:

1. Mr. Satya Ranjan Das, Faculty (CSE), ITER
Contact # satyadas@soa.ac.in
2. Mr. Paresh Baidya, Faculty (CSE), ITER
Contact # pareshbaidya@soa.ac.in
3. Dr. Lambodar Jena, Faculty (CSE), ITER
Contact # lambodarjena@soa.ac.in
4. Dr. Rajesh Purkait, Faculty (CSE), ITER
Contact #krishanumaity@soa.ac.in
5. Mr. Prithviraj Mohanty, Faculty (CS&IT), ITER
Contact # prithvirajmohanty@soa.ac.in
6. Mrs. Sarita Mahapatra, Faculty (CS&IT), ITER
Contact # saritamahapatra@soa.ac.in
7. Dr. Tribhuvan Singh, Faculty (CSE), ITER
Contact #tribhuvansingh@soa.ac.in
8. Mr. Naveen Kumar, Faculty (CSE), ITER
Contact # naveenkumar@soa.ac.in
9. Mr. Rangaballav Pradhan, Faculty (CSE), ITER
Contact #rangaballavpradhan@soa.ac.in
10. Mr. Suchintan Mishra, Faculty (CSE), ITER
Contact # suchintanmishra@soa.ac.in
11. Mrs. Aditi Panda, Faculty (CSE), ITER
Contact # aditipanda@soa.ac.in
12. Mr. M. Thangaval, Faculty (CSE), ITER
Contact #mthangavel@soa.ac.in

5. Text Book and References:

Text book

1. (T1) Algorithm Design by Jon Kleinberg and Eva Tardos, Pearson Publication

Reference book

1. (R1) The Algorithm Design Manual by Steven Skiena, Springer Publication
2. (R2) Introduction to Algorithms by CLRS, PHI Publication

6. Specific Course Information:

a. Course Description:

The course topics will include concepts of algorithm complexity, and various algorithmic design patterns like greedy approach, divide-and-conquer, and dynamic programming. 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 2031(INT), CSE 1002(DM), CSE 2001(DSA)
Co-requisite: CSE 2041(CSW1)

7. Course Learning Outcomes (CLOs):

By the end of course through lectures, readings, home-works, lab assignments and exams, students will be able to demonstrate the abilities:

(1) to apply knowledge of computing and mathematics to algorithm design;
(2) to argue/prove correctness of algorithms using inductive proofs and invariants;
(3) to analyze worst-case running times of algorithms using asymptotic analysis;
(4) to explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate.
(5) to describe the greedy paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize greedy algorithms. Derive and solve recurrences describing the performance of greedy algorithms.
(6) to describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize divide-and-conquer algorithms. Derive and solve recurrences describing the performance of divide-and-conquer algorithms.
(7) to describe the dynamic programming paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize dynamic programming algorithms. Derive and solve recurrences describing the performance of dynamic programming algorithms.

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

- Introduction to algorithm design
  - Algorithm correctness
  - Algorithm analysis
- Graphs and related algorithms
- Greedy approach
- Divide-and-conquer
- Dynamic Programming

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	**Introduction to Algorithm Design:** Importance of problem solving using algorithms; Characteristic features of an algorithm(input, output, finiteness, definiteness, effectiveness, correctness, efficiency);	
L 03	**Introduction to Algorithm Design:** Expressing algorithms (pseudocode); Basic aspects of algorithms (design and analysis)	
PSS 01	Stable Matching Problem; Five Representative Problems	To be referred from T1(1.1, 1.2)
Week # 2:		
L 04	**Algorithm Correctness:** using counter examples, loop invariants, induction method	To be referred from R1
L 05	**Algorithm Correctness:** using counter examples, loop invariants, induction method	
L 06	**Algorithm Correctness:** using counter examples, loop invariants, induction method	
PSS 02	Discussion on correctness of various recursive and iterative algorithms	To be referred from R1,R2
Week # 3:		
L 07	**Introduction to Algorithm Design:** Time and space complexity of an algorithm	To be referred from T1(Chapter 2)
L 08	**Basics of Algorithm Analysis:** Asymptotic notations; Summations; Logarithms	
L 09	**Basics of Algorithm Analysis:** Asymptotic notations; Summations; Logarithms (contd..)	
PSS 03	Discussion on exercise problems	To be referred from T1, R2
Week # 4:		
L 10	**Basics of Algorithm Analysis:** Recurrences	To be referred from R2
L 11	**Basics of Algorithm Analysis:** Recurrences (contd..)	
L 12	**Basics of Algorithm Analysis:** Recurrences (contd..)	
PSS 04	Discussion on exercise problems	To be referred from R1, R2
Week # 5:		
L 13	**Sorting and Searching:** Heap and Heap sort	To be referred from R2
L 14	**Sorting and Searching:** Heap and Heap sort	
L 15	**Sorting and Searching:** External and internal sorting; In-place sorting; Stable sorting; Special cases of sorting and searching etc.	

PSS 05	Discussion on numerical and analytical questions on sorting(bubble, insertion, selection, radix etc.) and searching (linear, binary)	
Week # 6:		
L 16	**Graph:** Basic definitions, applications and representations	To be referred from T1(Chapter 3) and R2(Chapter 22)
L 17	**Graph:** Basic definitions, applications and representations (contd..)	
L 18	**Graph:** Graph connectivity and graph traversal (BFS, DFS)	
PSS 06	Discussion on applications and variants of graph based algorithms already discussed	
Week # 7:		
L 19	**Graph:** Graph connectivity and graph traversal (BFS, DFS)	To be referred from T1(Chapter 3) and R2(Chapter 22)
L 20	**Graph:** Testing bipartiteness – an application of BFS	
L 21	**Graph:** Connectivity in directed graph; Directed-Acyclic-Graph and Topological ordering	
PSS 07	Discussion on applications and variants of graph based algorithms already discussed	
Week # 8:		
L 22	**Graph:** Connectivity in directed graph; Directed-Acyclic-Graph and Topological ordering	To be referred from T1(Chapter 3) and R2(Chapter 22, 23)
L 23	**Graph:** MST using Kruskal's algorithm—the union-find data structure	
L 24	**Graph:** MST using Kruskal's algorithm—the union-find data structure (contd..)	
PSS 08	Discussion on applications and variants of graph based algorithms already discussed	
Week # 9:		
L 25	**Graph:** MST using Prim's algorithm	To be referred from T1(Chapter 3) and R2(Chapter 23, 24)
L 26	**Graph:** Shortest path problem (Dijkstra' algorithm)	
L 27	**Graph:** Shortest path problem (Dijkstra' algorithm)	
PSS 09	Discussion on applications and variants of graph based algorithms already discussed	
Week # 10:		
L 28	**Greedy Method:** Interval Scheduling: The Greedy Algorithm Stays Ahead	To be referred from T1(Chapter 4)
L 29	**Greedy Method:** Scheduling to Minimize Lateness: An Exchange Argument	
L 30	**Greedy Method:** Optimal Caching: A More Complex Exchange Argument	
PSS 10	Discussion on variants of greedy method based problems already discussed	
Week # 11:		
L 31	**Greedy Method:** Huffman Codes and Data Compression	To be referred from T1(Chapter 4) and R2(Chapter 16)
L 32	**Greedy Method:** Huffman Codes and Data Compression (contd..)	
L 33	**Greedy Method:** Clustering	
PSS 11	Discussion on variants of greedy method based problems already discussed	
Week # 12:		
L 34	**Divide and Conquer:** Control abstraction; Merge sort	To be referred from T1(Chapter 5)
L 35	**Divide and Conquer:** Counting inversions	
L 36	**Divide and Conquer:** Quick sort	

PSS 12	Discussion on variants of divide and conquer based problems already discussed	
Week # 13:		
L 37	**Divide and Conquer:** Closest Pair of Points	To be referred from T1(Chapter 5)
L 38	**Divide and Conquer:** Karatsuba algorithm for fast integer multiplication	
L 39	**Divide and Conquer:** Convolutions and FFT	
PSS 13	Discussion on variants of divide and conquer based problems already discussed	
Week # 14:		
L 40	**Dynamic Programming:** Control abstraction; Recursion vs. Memoization	To be referred from T1(Chapter 6)
L 41	**Dynamic Programming:** Generating nth Fibonacci number	
L 42	**Dynamic Programming:** Computing binomial coefficient	
PSS 14	Discussion on variants of dynamic programming based problems already discussed	
Week # 15:		
L 43	**Dynamic Programming:** Matrix Chain Multiplication	To be referred from T1(Chapter 6)
L 44	**Dynamic Programming:** String Matching using Edit Distance	
L 45	**Dynamic Programming:** Longest Common Subsequence	
PSS 15	Discussion on variants of dynamic programming based problems already discussed	
Week # 16:		
L 46	**Revision class**	
L 47	**Revision class**	
L 48	**Revision class**	
PSS 16	**Quiz Test**	

9. Evaluation scheme (under GP1):

Assignments and Project**: 20%
Attendance: 5%
Mid-semester: 15%
End-semester(Lab. Test/Quiz): 15%
End-semester(Theory): 45%

** *Students have to submit soft copies as well as hard copies for the project on the last day of teaching.*