Course Profile:

Course Title: Algorithm Design and Analysis Laboratory
Credit Hour: 1.00 credits
Level/Term: Level 2 Term 2
Prerequisite: Structured Programming, Data Structure

Course Code: CSE 226
Conduct Hours: 13
Type: Core/Major: Core

Rationale:
Intended to enable the learners to analyze and criticize the complexity of existing bench-mark algorithms, use the acquired knowledge to understand advanced algorithms, and develop or design efficient algorithms to solve real world problems.

Course Objectives:

Upon completion of this course, students will be able to do the following:

1. Demonstrate a familiarity with major algorithms and data structures.
2. Apply important algorithmic design paradigms and methods of analysis.
3. Analyze the asymptotic performance of algorithms.
4. Prove rigorous correctness for algorithms.
5. Synthesize efficient algorithms in common engineering design situations.

Course Description (Catalog Description):

This course introduces basic methods for the design and analysis of efficient algorithms emphasizing methods useful in practice. Different algorithms for a given computational task are presented and their relative merits evaluated based on performance measures. The following important computational problems will be discussed: sorting, searching, elements of dynamic programming and greedy algorithms, advanced data structures, graph algorithms (shortest path, spanning trees, and tree traversals), string matching, elements of computational geometry, NP completeness.

Course Outcomes (COs): By the end of the course, students will be able to do the followings:

- 1: apply Standard Template Library and JAVA collection
- 2: compute the space and time complexity of an algorithm
- 3: apply knowledge from an algorithm to solve other real-world problems
- 4: analyze different algorithms for solving a problem

Text and Reference books:
Text Books:

1. Introduction to Algorithms **(Third Edition),** by Coreman *et* al., Publishers: McGraw-Hill

References:

1. Fundamentals of Computer Algorithms, Sartaj Sahni and Sanguthevar Rajasekaran Ellis Horowitz

Teaching Strategy: Typical methodologies are Class lectures, web-access, self-study, critics writing, problem formulation, and student presentation.

Assessment Strategy:

Description	Marks
Class Attendance/ Participation	10
Performance	10
Reports	20
Viva	10
Lab Exam	50

CO Delivery and Assessment:

Cos	Corresponding POs	Bloom's taxonomy domain/level *(C: Cognitive, P: Psychomotor A: Affective)*	Delivery Methods and activities	Assessment tools
CO1	PO1, PO5	C3	Lecture, Web	Lab Performance, Lab Exam,Quiz
CO2	P01, PO2, PO3	C6	Lecture, Web	Lab Performance, Lab Exam,Quiz
CO3	PO1, PO2, PO3, PO4	C3	Lecture, Web	Lab Performance, Lab Exam
CO4	PO1, PO2, PO3, PO4	C4	Lecture, Web	Lab Performance, Lab Exam

(Recall:
2. Domains and Levels of Bloom's Taxonomy
□ "Cognitive" Domain (C): C1 - Recall data, C2 - Understand, C3 - Apply, C4 - Analysis, C5 - Synthesize, and C6 - Evaluate.
□ "Affective" Domain (A): A1 - Receive, A2 - Respond, A3 - Value, A4 - Organize personal value system, and A5 - Internalize value system.
□ "Psychomotor" Domain (P): P1 - Imitation, P2 - Manipulation, P3 - Develop precision, P4 -Articulation, and P5 - Naturalization.)

CO-PO Mapping (Theory course):

CO/PO mapping (S/M/W indicates strength of correlation) S-Strong, M-Medium, W- Weak												
Cos	Program Outcomes (POs)											
	PO 1	PO 2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	√				√							
CO2	√	√	√									
CO3	√	√	√	√								
CO4	√	√	√	√								

Department of Computer Science and Engineering
Lesson Plan:

Course Title: Algorithm Design & Analysis Lab
Level/Term: Level-2 Term-2
Credit: 01
Prerequisite: Structured Programming, Data Structure
Type: Core/Major:Core
Session: February 2019

Course Code: CSE-226
Section: A , B
Conduct Hours: 13

Instructor: Syed Md. Minhaz Hossain
Class schedule:

Email address: minhazpucese@gmail.com

Room No: 502
Phone No: 01815811328

Rationale:

Intended to enable the learners to analyze and criticize the complexity of existing bench-mark algorithms, use the acquired knowledge to understand advanced algorithms, and develop or design efficient algorithms to solve real world problems.

Course Objectives:

Upon completion of this course, students will be able to do the following:

1. Demonstrate a familiarity with major algorithms and data structures.
2. Apply important algorithmic design paradigms and methods of analysis.
3. Analyze the asymptotic performance of algorithms.
4. Prove rigorous correctness for algorithms.
5. Synthesize efficient algorithms in common engineering design situations.

Course Outcomes (COs):
Students who complete the course will have demonstrated the ability to do the following:

1. Pick the appropriate data structures.
2. Apply the appropriate algorithms in different situations (randomized, iterative, recursive, dynamic, greedy, graph, backtracking, branch and bound).
3. Test efficiency of algorithm (asymptotic notation)
4. Argue the correctness of algorithms.

Marks Distribution:

Description	Marks
Class Attendance/ Participation	10
Performance	10
Reports	20

Viva	10
Lab Exam	50

Weekly schedule:

Week/Class	Topic	Teaching strategy	Course outcome	Assessment Strategy
Class -1	Using Open, implement a parallelized Merge Sort algorithm to sort a given set of elements and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.	Lab Manual , Video Tutorial	CO1,CO2,CO4	Problem Solving
Class-2	Sort a given set of elements using the Quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the llst to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.	Lab Manual , Video Tutorial	CO1,CO2,CO4	Problem Solving
Class 3	Implement 0/1 Knapsack problem using Dynamic and Fractional Knapsack using Greedy Method.	Lab Manual , Video Tutorial	CO3,CO4	Problem Solving
Class-4	Find a subset of a given set S = {sl,s2,.....,sn} of n positive integers whose sum is equal to a given positive integer d. For example, if S= {1, 2, 5, 6, 8} and d = 9 there are two solutions{1,2,6}and{1,8}.A suitable message is to be displayed if the given problem instance doesn't have a solution.	Lab Manual , Video Tutorial	CO3,CO4	Problem Solving
Class-5	a. Print all the nodes reachable from a given starting node in a digraph using BFS method. b. Check whether a given graph is connected or not using DFS method.	Lab Manual , Video Tutorial	CO1,CO4	Problem Solving

Class-6	Find Minimum Cost Spanning Tree of a given undirected graph using Prim□s algorithm.	Lab Manual , Video Tutorial	CO1,CO4	Problem Solving
Class-7	Find Minimum Cost Spanning Tree of a given undirected graph using Krushkal□s algorithm.	Lab Manual , Video Tutorial	CO1,CO4	Problem Solving
Class-8	From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijikstra□s algorithm.	Lab Manual , Video Tutorial	CO1,CO4	Problem Solving
Class-9	a. Obtain the Topological ordering of vertices in a given digraph. b.Compute the transitive closure of a given directed graph using Warshall's algorithm.	Lab Manual , Video Tutorial	CO1,CO4	Problem Solving
Class-10	Implement any scheme to find the optimal solution for the Traveling Salesperson problem and then solve the same problem instance using any approximation algorithm and determine the error in the approximation.	Lab Manual , Video Tutorial	CO1,CO2,CO3,CO4	Problem Solving
Class-11	Design and implement the presence of **Hamiltonian Cycle** in an undirected Graph **G** of ***n*** vertices.	Lab Manual , Video Tutorial	CO1,CO3,CO4	Problem Solving
Class-12	Implement N Queen's problem using Back Tracking.	Lab Manual , Video Tutorial	CO1,CO3,CO4	Problem Solving
Class-13	Lab Exam			Quiz, Problem Solve ,Viva