CMPE 371 Analysis of Algorithms

Department:
Computer Engineering

Program Name: Computer Engineering	**Program Code:** 25	
Course Number: CMPE371	**Credits:** 4 Cr	**Year/Semester:** 2023-2024 FALL

☒ Required Course ☐ Elective Course (click on and check the appropriate box)

Prerequisite(s):
CMPE231

Catalog Description:
Definition and properties of Algorithms. Design, analysis, and representation of Algorithms. Data abstraction. Pseudo code conventions. Models of computation. Mathematical Foundations: Growth of functions, asymptotic notations. Study of recursive algorithms and associated recurrence relations (substitution method, iteration method, master method, recursion trees). Design paradigms for algorithms: Brute-Force (Exhaustive Search), Divide-and-Conquer (Merge Sort, Binary Search Tree) Dynamic Programming (Matrix-Chain multiplication, LCS-length, 01-Knapsack Problem). Greedy algorithms (Greedy Activity Selector, Fractional Knapsack Problem). Graph Algorithms: Representation of sets and graphs. Breadth-first search, depth-first search..

Course Web Page:
https://staff.emu.edu.tr/ahmetunveren/en/teaching/cmpe371/

Textbook(s):
Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, "Introduction to ALGORITHMS", MIT Press.
Lab Manual:
NA

Indicative Basic Reading List :
Anany Levitin "Introduction to Design and Analysis of Algorithms", Addvison Wesley, 2003

Topics Covered and Class Schedule:
(4 hours of lectures per week)

Week 1-2	Definition and properties of Algorithms. Design, analysis, and representation of Algorithms. Data abstraction. Pseudo code conventions. Models of computation.
Week 3-4	Mathematical Foundations: Growth of functions, asymptotic notations.
Week 5-6-7	Study of recursive algorithms and associated recurrence relations (substitution method, iteration method, master method, recursion trees).
Week 8-9	**Midterm Exam**
Week 10-11	Brute-Force (Exhaustive Search), Divide-and-Conquer (Merge Sort, Binary Search Tree).
Week 12-13	Dynamic Programming (Matrix-Chain Multiplication, LCS-length, 0-1 Knapsack Problem).
Week 13	**Midterm Exam**
Week 14	Greedy algorithms (Greedy Activity Selector, Fractional Knapsack Problem).
Weeks 15	Graph Algorithms: Representation of sets and graphs. Breadth-first search, depth-first search.
Week 16-17-18	**Final Examination**

Tutorials/Laboratory Schedule:
(2 hours of laboratory per week)

There will be regular tutorial/lab lectures. Schedules will be announced in class.

Course Learning Outcomes:

At the end of the course, student must be able to

1. Find time complexity of an algorithm.
They can use asymptotic notations while comparing time complexity of algorithms. Can formulate the recurrence relation for the algorithm and solve it.

2. Carry out a complete algorithmic design process (design, analysis, implementation, results).
They can solve problems involving algorithmic design, analysis, and implementation.

3. Analyze the given problem and solve it by using appropriate programming technique.
They can use Brute-Force, Divide-and-Conquer, Dynamic Programming and Greedy algorithms for the solution of the given problem.

4. Possess the mathematical knowledge and skills necessary to the analysis of algorithms:Reinforce mathematical fundamentals including techniques for solving summations and recurrences and the asymptotic growth rate of functions.

5. Gain insight into algorithmic design: The mechanisms that affect algorithmic logic, structure, and performance.

6. Demonstrate their ability to carry out a complete algorithmic design process (design, analysis, implementation, results).

7. Gain an understanding of certain classes of algorithms, along with models for future algorithmic work.

8. Introduce a number of standard algorithms, both classical and modern, as objects for algorithmic analysis.

9. Address problems involving algorithmic design, analysis, and implementation.

10. Apply proof techniques and mathematical concepts to demonstrate the correctness and assess the performance of standard algorithms

	Method	**#**	**Percentage**
Assessment	Midterm 1	1	25%
	Midterm 2	1	30%
	Lab Work(s): Tutorials	10	5 %
	Final Examination	1	45%

Contribution of Course to Criterion 5

Credit Hours for:
Mathematics & Basic Science : 0
Engineering Sciences and Design : 4
General Education : 0

Relationship of Course to Program Outcomes

The course supports achievement of the following program objectives

- I. Identify, formulate and solve computer engineering and science problems ...
- VII. Apply modern engineering tools and techniques innovatively;
- X. Pursue graduate studies in related fields.

This course is used to assess the following items of Program Outcomes

- 1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.
- 6) an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions

Prepared by: Asst. Prof. Dr. Ahmet Ünveren (C) Assoc. Prof. Dr. Adnan Acan	**Date Prepared:** September 2023