Course number and name: CS 07340: Design and Analysis of Algorithms
Credits and contact hours: 3 credits / 3 contact hours
Instructor’s or course coordinator’s name: Andrea Lobo

Specific course information

Catalog description: In this course, students will learn to design and analyze efficient algorithms for sorting, searching, graphs, sets, matrices, and other applications. Students will also learn to recognize and prove NP-Completeness.

Prerequisites: CS07210 Foundations of Computer Science and CS04222 Data Structures and Algorithms

Type of Course: ☒ Required ☐ Elective ☐ Selected Elective

Specific goals for the course

1. **algorithm complexity.** Students have analyzed the worst-case runtime complexity of algorithms including the quantification of resources required for computation of basic problems.
   - ABET (j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices

2. **algorithm design.** Students have applied multiple algorithm design strategies.
   - ABET (j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices

3. **classic algorithms.** Students have demonstrated understanding of algorithms for several well-known computer science problems
   - ABET (j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices
design choices and are able to implement these algorithms.

4. **NP complete.** Students have written NP-completeness proofs.

   - ABET (j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices

Required list of topics to be covered

1. Brute Force and Exhaustive Search
2. Mathematical preliminaries
3. Worst-case algorithm analysis
4. Recurrence relations and analysis of recursive algorithms
5. Divide and Conquer
6. Dynamic Programming
7. Greedy algorithm approach
8. Backtracking
9. Branch and Bound algorithm design
10. Advanced Data Structures: Graphs, Union-Find
11. NP-Completeness

Optional list of topics that could be covered

12. Approximation algorithms
13. Randomized algorithms
14. Balanced trees, heaps