Credits and contact hours: Credit hours: 3, Contact Hours: 42 (1 contact hour = 50 minutes)

Instructor: Dr. Jodi Tims


Specific course information
a. Catalog description: Building on concepts of basic algorithm design and complexity analysis introduced in earlier courses, this course explores advanced algorithm design and analysis. Topics may include advanced data structures, inductive algorithms, graph algorithms, geometric, algebraic, and numeric algorithms, reductions, NP-completeness, and parallel algorithms. Emphasis is placed on formal efficiency analysis of algorithms utilizing concepts from discrete mathematics.

b. Prerequisites: CSC-245

c. Required/Elective:
i. Computer Science – required
   ii. Software Engineering - required

Specific goals for the course
a. Specific outcomes of instruction
   i. The student will be able to explain why a study of algorithms is important to an overall understanding of computer science.
   
ii. The student will be able to describe various problem solving strategies such as brute force, divide and conquer, dynamic programming, greedy algorithms, backtracking, and branch and bound.
   
iii. The student will be able to explain how the choice of data structures impacts the ability to solve a problem and/or to increase the efficiency of a problem’s solution.
   
iv. The student will be able to apply problem solving techniques to a wide variety of problems including graph analysis, scheduling, string processing, geometric problems, and sorting and searching.

v. The student will be able to formally analyze the efficiency of an algorithm using recurrence relations and asymptotic notations.

vi. The student will understand that not all problems can be efficiently solved via algorithmic means.

vii. The student will be able to describe how approximations to problems solutions in the form of heuristic algorithms may be an effective approach for handling some problems.

a. CAC Criterion 3 outcomes addressed by the course:
a. An ability to apply knowledge of computing and mathematics appropriate to the program’s student outcomes and to the discipline
f. An ability to communicate effectively with a range of audiences.
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.

b. EAC Criterion 3 outcomes addressed by the course:
a. An ability to apply knowledge of mathematics, science, and engineering
e. An ability to identify, formulate, and solve engineering problems
i. A recognition of the need for, and an ability to engage in life-long learning.
k. An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

Brief list of topics to be covered
a. Course introduction and review of basic data structures (3 class periods)
b. Recurrence relations (3 class periods)
c. Brute force (3 class periods)
d. Decrease and conquer (3 class periods)
e. Divide and conquer (4.5 class periods)
f. Transform and conquer (1 class period)
g. Space-Time tradeoffs (1.5 class period)
h. Dynamic programming (4.5 class periods)
i. Greedy Algorithms (4.5 class periods)
j. Union-find data structure (1.5 class periods)
k. Iterative improvement (3 class periods)
l. Limitations of algorithmic power (1.5 class periods)
m. Backtracking (1.5 class periods)
n. Branch and Bound (1.5 class periods)
o. Exams (5 class periods)