THE CATHOLIC UNIVERSITY of AMERICA

Department of Computer Science

CSC411 Design and Analysis of Algorithms

Spring 2026

Instructor:	Dr. Richard Kelley
E-mail:	kelleyrc@cua.edu
Classroom:	McCort Ward 209
Day/Time:	Tue/Thu 2:10 PM - 3:25 PM
Office Hours:	Wed 10:00 AM - 11:00 AM By appointment

Course Description

This course presents the fundamental techniques for designing efficient computer algorithms, providing correctness, and analyzing the complexity of algorithms. General topics include methods for expressing and comparing the complexity of algorithms: worst and average cases, sorting, selection, graph algorithms, and basic algorithm design techniques such as divide-and-conquer, greedy method, backtracking, and dynamic programming.

Prerequisites

Courses: CSC 280 – Data Structure

Textbooks

Introduction to Algorithms, 4th Edition
Author: Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein
Publisher: The MIT Press; 4th edition (April 5, 2022)
https://www.amazon.com/Introduction-Algorithms-fourth-Thomas-Cormen/dp/026204630X

Introduction to the Design and Analysis of Algorithms, 3rd Edition
Author: Anany V. Levitin
Publisher: Addison Wesley; 3rd edition (October 9, 2011)
https://www.amazon.com/Introduction-Design-Analysis-Algorithms-3rd/dp/0132316811/

Course Objectives

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

- Analyze asymptotic runtime complexity of algorithms.
- Demonstrate a familiarity with major algorithms and data structures.
- Understand and design algorithms using greedy strategy, divide and conquer approach, and dynamic programming etc.
- Apply important algorithmic design paradigms and methods of analysis.

Grading

Grades will be based on one midterm exam (20%), one final exam (20%), quizzes (20%), homework (20%), in-class exercise (10%), and class participation/attendance (10%).

- **Examinations (40%) -** Midterm and final examinations are closed book.
- **Quizzes (20%) -** Quizzes will be given weekly at the beginning of a class (usually on Tuesday). A quiz usually takes about 10 minutes and contains 1~3 questions from the previous lecture(s).
- **Homework (20%) -** Homework is important and can be challenging in this course. Working on your homework is critical to doing well on the midterm and final exams. The homework assignments will be graded based on the correctness and your efforts in solving the assigned problems.
- **In-class Exercise and Class Participation (20%) -** Requirements include class attendance, active participation in discussions, and in-class problem solving. Please let the instructor know in advance if you expect to be absent for any reason. If you must miss a class due to an emergency or documented illness, email the instructor as soon as possible. You are still responsible for any material covered, assignments given, and homework due during the missed classes unless pre-approved by the instructor.
- Point totals will convert to letter grades according to the following table. These intervals are closed on the left and open on the right (so earning 95% of the total possible points in the class will get an A, but (95-x)% will earn an A- for any positive value of x).

Score	Grade
95%	A
90%-95%	A-
87%-90%	B+
83%-87%	B
80%-83%	B-
77%-80%	C+
73%-77%	C
70%-73%	C-
66%-70%	D
0%-66%	F

- Remember: *grades in Brightspace do not necessarily forecast the final course grade, as they may not reflect outstanding assignments.*

The University grading system is available:

- Undergraduate policies on grades and academic standing
- Graduate policies on grades and academic standing

Reports of grades in courses are available at the end of each term in Cardinal Students.

Topics Covered

1. Fundamentals of the Analysis of Algorithm Efficiency

2. Review of Fundamental Data Structure
3. Big-O Notation
4. Algorithm Design Techniques
5. Brute Force and Exhaustive Search
6. Greedy Technique
7. Divide-and-Conquer
8. Back Tracking
9. Dynamic Programming

Tentative Class Schedule

Schedule is subject to change. Additional details can be found on the course webpage: https://richardkelley.io/CSC411.

- Week 1: Algorithmic Problem Solving and Common Problem Types
- Week 2: Fundamentals of Algorithm Analysis
- Week 3: Asymptotic Analysis and Math Review
- Week 4: Exhaustive Search
- Week 5: Recursive Problems and Solutions
- Week 6: Divide and Conquer
- Week 7: Transformations and Reductions
- Week 8: Midterm exam.
- Week 9: Spring Break
- Week 10: Tradeoffs Between Space and Time Complexity
- Week 11-12: Dynamic Programming
- Week 13: Greedy Algorithms
- Week 14: Complexity Lower Bounds
- Week 15: Approximation Algorithms
- Week 16: Final Exam Review

Expected Course Outcomes

Students at the end of the course should have a clear understanding of the following concepts and topics:

1. Learn good principles of algorithm design
2. Understand the limiting factors of resources in algorithm design (such as time and space in algorithmic solutions)
3. Learn basic algorithm design methods (such as the greedy technique, divide-and-conquer, and backtracking)
4. Learn how to analyze the performance of algorithms, i.e., learn how to analyze algorithms and estimate their worst-case and average-case behaviors, and estimate the time complexity using the big-O notation
5. Understand how to approach the algorithm design and analysis, i.e., which principles and/or design methods to use for a giving problem; how to compare the quality of different algorithms.

ABET Student Outcomes Addressed by this Course

SO1. An ability to analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.

SO6. Apply computer science theory and software development fundamentals to produce computing-based solutions.

Outcome Assessment

The course employs the following mechanisms to assess the above learning outcomes:

1. Homework is assigned and graded to assess the level of student understanding of course topics. The learning outcomes are also assessed through quizzes, mid-term and final exams given during the semester.
2. The classroom questions (in-class exercises) are used by the instructor to check the degree of students' understanding on the subjects.
3. The overall assessment of the course is done through the University student course evaluation.

Process of Improvement

The instructor continuously tries to improve the course as described below:

1. The instructor evaluates students' performance through in-class questions, homework, quizzes and examinations.
2. The instructor carefully examines the suggestions made by students during the semester, and may make necessary adjustment on class schedule.
3. The university conducts a formal course evaluation at the end of each semester. The results of the evaluation are used to assess various aspects of effectiveness of learning in the course.
4. At the end of every calendar year, the EECS faculty will conduct ABET course review on the course. The evaluation result and recommendations (if any) will be provided to the instructor for continuous improvement on the course.

Contribution of Course to Professional Component

This course builds the computer science foundation in the area of programming. Students learn how to analyze the complexity of algorithms and how to design computer algorithms using proper methods when dealing with different real-world problems.

UNIVERSITY POLICIES

All of Catholic University's policies are detailed at Catholic University Policy Webpage.

Academic Integrity

Academic dishonesty at The Catholic University of America is not tolerated. As such, academic integrity is not merely avoiding plagiarism or cheating, but it certainly includes those things. Academic integrity means, above all else, taking responsibility for your work, your ideas, and your effort, and giving credit to others for their work, ideas, and effort. If you submit work that is not your own – whether test answers, whole papers, or something in-between – that is considered to be academic dishonesty. University procedures related to academic dishonesty are conducted with respect and dignity, while also preserving accountability, and they presuppose that all participants will treat each other with respect and dignity.

- Undergraduate Student Academic Dishonesty Policy
- Graduate Student Academic Dishonesty Policy

Grades and Academic Standing

- Undergraduate policies on grades and academic standing
- Graduate policies on grades and academic standing

University Recording Policies

- Recording Classroom Lectures Policy
- CUA Recording Policy

Accommodations for students with disabilities

Any student who feels they may need a reasonable accommodation based on the impact of a disability should contact the Office of Disability Support Services (Office of Disability Support Services) by emailing at CUA-DSS@cua.edu