CSUDH Computer Science Department

CSC401: Analysis of Algorithms
CSC501: Advanced Algorithm Analysis and Design
Fall 2022

Instructor:	Dr. Jianchao (Jack) Han		
Phone number:	310-243-2624	**Classroom:**	SAC 2104
Office:	NSM A-133	**Office Hours:**	Mondays 5pm-7pm
Email:	jhan@csudh.edu		

Prerequisites: CSC123, CSC311, CSC281/MAT281, Math 321
Textbook: Introduction to the Design & Analysis of Algorithms, 3rd. Ed. By Anany Levitin, Addison Wesley, 2012, ISBN-13: 978-0-13-231681-1.

Course Contents:

The course will cover the following contents:

- *Methodologies for algorithm analysis*: experimental analysis and asymptotic analysis of algorithm running time efficiency and storage requirement
- *Algorithm analysis preparation*: pseudo code, primitive instructions counting, preliminary mathematics
- *Basic data structures and related algorithms' analysis*: stacks, queues, priority queues, heaps, dictionaries, hash tables, trees, binary trees, multiple trees, balanced trees, etc.
- *Efficiency analysis techniques*: methods for determining the efficiency of an algorithm and for determining the best possible efficiency for particular problems.
- *Brute-force algorithms*: these algorithms are based on a "try everything and see what works" approach.
- *Divide and conquer algorithms*: these algorithms first solve smaller instances of a problem, then combine the solutions to solve the complete problem.
- *Decrease and conquer algorithms*: these algorithms exploit the relationship between a problem and a smaller instance of the same problem.
- *Transform and conquer algorithms*: these algorithms work by transforming a problem into a problem that is more readily solvable or already solved.
- *Space and time trade-offs*: these algorithms use more memory to yield more time-efficient solutions.
- *Dynamic Programming*: these algorithms are used to solve problems that have overlapping sub-problems by solving each sub-problem once and recording the solutions in a manner that can be used to construct a solution to the complete problem.
- *Greedy algorithms*: these algorithms make decisions based on what appears to be the best next step at each stage in the solution process.
- *Iterative improvement algorithms*: these algorithms start with a feasible solution and then improve it in small steps until reaching a solution that cannot be improved.
- *Very hard problems*: Some problems cannot be solved efficiently by any algorithm. These are the well-known P, NP and NP-complete classes of problems.

- Tackling very hard problem algorithms: these are strategies for dealing with very hard problems, including approximation algorithms, backtracking algorithms, branch-and-bound algorithms

Course Objectives: By the end of this course, the students should be able to use fundamental algorithm analysis and design techniques to solve real-life problems. Specifically, the students should

- Master the methodologies of algorithm analysis
- Completely understand the basic data structures and related algorithms
- Completely understand binary search trees and balance maintenance
- Have a basic understanding of lower-bound on comparison-based sorting algorithms and other linear sorting algorithms
- Understand various algorithm design and analysis techniques: brute-force, the greedy method, divide-and-conquer, and dynamic programming approaches and be able to use these techniques to design your own algorithms
- Understand graph representation data structures and be able to use related typical algorithms to solve real-life problems
- Have a basic understanding of text processing, web search engine, text pattern matching, text compression, and gene alignment algorithms
- Be able to transform generic algorithmic description to working programs to solve specific problems
- Understand P and NP problems, be familiar with NP-Completeness and typical NP-Complete problems, and be able to use approximation, backtracking, and branch-and-bound approaches to solve NP-Complete problems

Requirements: To pass this course, you must meet the following requirements:

- **Individual work:** Students must complete all tests and quizzes individually.
  - **Tests:** There will be **ONE** midterm test and **ONE** final test, both of which will be comprehensive and will cover the contents contained in the textbook and our course notes.
  - **Quizzes: Twelve quizzes** will be given, one for each chapter. All quiz questions are multiple-choice questions and each quiz has four questions.

- **Group work:** Both written assignments and programming projects will be group work. Each group will be 3 or 4 students, and graduate students and undergraduate students should be in different groups.
  - **Assignments: Six** group written assignments, each of which will have two weeks for you to complete. Each assignment will have some extra questions for graduate students only.
  - **Programming Projects: Four** group programming projects will be scheduled evenly during the session, each of which will have about two weeks to complete.

- **Graduate research project:** Each graduate student must complete a research project. The research topic should be related to algorithm design and analysis and approved by the instructor.

Grading Breakdown:

- Undergraduate students (CSC401)
  - 12 Quizzes: 12%, 1% each
  - 6 Written Assignments: 24%, 4% each
  - 4 Programming Projects: 20%, 5% each
  - 1 Midterm Test: 16%
  - 1 Final Test: 28%
- Graduate students (CSC501)
  - Undergraduate student requirement 85%
  - Research project 15%

Grading Scale:

[93,100] = A	[90-93) = A-	
[85-90) = B+	[80-85) = B	[75-80) = B-
[70-75) = C+	[65-70) = C	[60-65) = C-
[55-60) = D+	[50-55) = D	below 50 = F

Tentative Class Schedule (subject to change): We will do our best to adhere to the following schedule. If any changes are necessary, you will be notified in class and in our course web page. You are always expected and encouraged to have read the appropriate sections of the textbook before coming to class.

Week	Topic	Ch.	Assignments	Quizzes	Projects
1 (8/29)	Introduction	1	A1	Q1	
2 (9/5)	**Labor Day (no class)**				
3 (9/12)	Analysis of algorithm efficiency	2		Q2	P1
4 (9/19)	Brute force algorithms	3	A2	Q3	
5 (9/26)	Divide and conquer	4		Q4	P2
6 (10/3)	Decrease and conquer	5	A3	Q5	
7 (10/10)	Transform and conquer	6		Q6	
8 (10/17)	**Midterm Test**	1-6			
9 (10/24)	Space and time trade-offs	7	A4	Q7	
10 (10/31)	Dynamic programming	8		Q8	P3
11 (11/7)	Greedy techniques	9	A5	Q9	
12 (11/14)	Iterative improvement	10		Q10	P4
13 (11/21)	Limitations of algorithm power	11	A6	Q11	
14 (11/28)	Coping with the limitations of algorithm power	12		Q12	
15 (12/5)	Graduates projects				
16 (12/12)	**Final Test**	1-12			

GENERAL POLICIES:

ACADEMIC HONOR CODE

Programming assignments must be done individually. Failure to do so will result in a violation of the CSUDH Academic Honor Code. The following cases will be considered as violations: identical code, and extremely similar code. Violations will be reported to the Office of Vice President of Academic Affairs.

STUDENT ACADEMIC APPEALS PROCESS

Authority and responsibility for assigning grades to student rests with the faculty. However, in those instances where students believe that miscommunication, error, or unfairness of any kind may have adversely affected the instructor's assessment of their academic performance, the student has a right to appeal by the procedure listed in the Undergraduate Catalog and by doing so within thirty days of receiving the grade or experiencing any other problematic academic event that prompted the complaint.

ADA STATEMENT

Students with disabilities, who believe they may need an academic adjustment in this class, are encouraged to contact me as soon as possible to better ensure receipt of timely adjustments.

DEFINITION OF CHEATING AND PLAGIARISM

CSUDH is dedicated to a high standard of academic integrity among its faculty and students. In becoming part of the California State University academic community, students are responsible for honesty and independent effort. Disciplinary action will be taken against any student who alone or with others engages in any act of academic fraud or deceit. (Read University Regulations in University Catalog).