

California State University Stanislaus  
Department of Computer Science  
Syllabus

**Instructor: Dr. Xuejun Liang**

My Office: DBH 282

Office Hours: MWF 10:00 a.m. - 11:00 a.m.

ZOOM Meeting ID: 4438930033, Phone: (209) 667-3169, Email: [xliang@csustan.edu](mailto:xliang@csustan.edu)

**Class Information**

Classroom: DBH 165

Class Date & Time: MWF 2:00 p.m. - 2:50 p.m.

Class Website: <https://www.cs.csustan.edu/~xliang/Courses3/CS4440-26S-S2>

Class Canvas: Use your class Canvas account to submit homework assignments

Class Modality: In-Person. Student attendance is required

**Catalog Description**

CS4440 Theory of Algorithms. (3 Hours) Pre-requisites: CS 3100 and MATH 2300. Students will learn fundamentals and various techniques of designing and analyzing computer algorithms. Topics include examples of computational problems, efficiency and correctness of algorithms, limitations of algorithm power, and fundamental techniques of algorithm designs such as brute force, divide-and-conquer, decrease-and-conquer, transform-and-conquer, dynamic programming, greedy, iterative improvement, and approximation algorithm design strategies.

**Textbook:**

A. Levitin, "[Introduction to the Design & Analysis of Algorithms](#)," 3rd Edition, Addison-Wesley, 2012, ISBN: 0132316811

**Reference Books:**

Jon Kleinberg and Eva Tardos, "[Algorithm Design](#)", Addison Wesley, 2006, ISBN: 9780321295354

**Course Outcomes**

Students who successfully complete the course should be able to

1. Analyze the time complexity of recursive and non-recursive algorithms with respect to the asymptotic order of growth.
2. Design and analyze algorithms to solve problems using brute-force and decrease-and-conquer techniques.
3. Design and analyze algorithms to solve problems using divide-and-conquer and transform-and-conquer techniques.
4. Understand space and time trade-offs of algorithm efficiency as well as design and analyze algorithms to solve problems using hashing and dynamic programming techniques.
5. Understand concept of suboptimal and satisfactory algorithms as well as design and analyze algorithms to solve problems using greedy and iterative improvement techniques.

6. Discuss lower bound arguments, decision tree, and NP-complete problem as well as design and analyze polynomial-time heuristics to approximate solutions for NP-complete problems.

### Course Outline\* (Major Topics and Weekly Schedule)

| Date                               | Topics Covered                                                                                                                                         | Sections       |
|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| Week 1:<br>01/28, 01/30            | Course syllabus. Introduction of the course.<br>Analysis framework; Asymptotic notations                                                               | 1.1-4<br>2.1-2 |
| Week 2:<br>02/02, 02/04,<br>02/06  | Mathematical analysis of nonrecursive algorithms<br>Mathematical analysis of recursive algorithms                                                      | 2.3-5<br>App.B |
| Week 3:<br>02/09, 02/11,<br>02/13  | Brute-force algorithms: Selection Sort, Sequential Search<br>Exhaustive search: Traveling-Salesman, Knapsack Problem                                   | 3.1-4          |
| Week 4:<br>02/16, 02/18,<br>02/20  | Depth-first search and breadth-first search<br><b>Test #1 (Chapter 1 to 3)</b><br>Decrease-by-one: insertion sort, topological sorting                 | 3.5<br>4.1-2   |
| Week 5:<br>02/23, 02/25,<br>02/27  | Algorithms for generating combinatorial objects, Binary search and other decrease-by-a-constant factor algorithms<br>Variable-size-decrease algorithms | 4.3-5          |
| Week 6:<br>03/02, 03/04,<br>03/06  | Divide-and-conquer: mergesort, quicksort<br>Other divide-and-conquer examples                                                                          | 5.1-2<br>5.4   |
| Week 7:<br>03/09, 03/11,<br>03/13  | Instance simplification: presorting, Gaussian elimination, balanced search trees                                                                       | 6.1-3          |
| Week 8:<br>03/16, 03/18,<br>03/20  | Representation change: heaps and heapsort, Horner's rule and binary exponentiation, and Problem reduction<br><b>Test #2 (Chapter 4 to 6)</b>           | 6.4-6          |
| Week 9:<br>03/23, 03/25,<br>03/27  | Space-time trade-offs: sort by counting, string matching, hashing, Btrees                                                                              | 7.1-4          |
| Week 10:<br>03/30, 04/01,<br>04/03 | Dynamic programming algorithms: Three basic examples, Knapsack Problem                                                                                 | 8.1-2          |
|                                    | <b>Spring Break</b>                                                                                                                                    |                |
| Week 11:<br>04/13, 04/15,<br>04/17 | Dynamic programming algorithms: Optimal Binary Search Trees, Warshall's, Floyd's<br><b>Test #3 (Chapter 7 to 8)</b>                                    | 8.3-4          |
| Week 12:<br>04/20, 04/22,<br>04/24 | Greedy algorithms: Prim's, Kruskal's, Dijkstra's,                                                                                                      | 9.1-3          |
| Week 13:<br>04/27, 04/29,<br>05/01 | Iterative improvement algorithms: Linear Programming, Maximum Flow Problem, Bipartite Graphs                                                           | 10.1-3         |

|                                    |                                                                                                                                                                                                                           |        |
|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| Week 14:<br>05/04, 05/06,<br>05/08 | Iterative improvement algorithms: Stable Marriage Problem<br><b>Test #4 (Chapter 9 to 10)</b>                                                                                                                             | 10.4   |
| Week 15:<br>05/11, 05/13,<br>05/15 | Lower-bound arguments, Decision trees, P, NP, and NP-complete problems<br>Review for the Final Exam.                                                                                                                      | 11.1-3 |
| <b>Week 16:</b><br><b>05/18:</b>   | <b>Final Examination: 05/18: 11:15 a.m.-1:15 p.m.</b><br><b>Final Examination Schedule</b><br><a href="https://www.csustan.edu/class-schedule/finals-schedule">https://www.csustan.edu/class-schedule/finals-schedule</a> |        |

\*It is subject to change.

## Grading Scale

Grading Scale will be assigned on a standard scale as below. Clustering of grades may cause the grading scale to be lowered (to your benefit), but it will not be raised.

| A      | B     | C     | D     | F   |
|--------|-------|-------|-------|-----|
| 90-100 | 75-89 | 60-74 | 45-59 | <45 |

## Evaluation:

The overall course grade will be the weighted sum of the points earned in the following categories:

| Attendance | Homework | Tests | Final Exam |
|------------|----------|-------|------------|
| 10%        | 20%      | 40%   | 30%        |

## Other Policies:

1. I will accept the homework assignments late for maximum three days (including holidays) with the point deduction 20% per day.
2. There will be no makeup tests except in a verified emergency with immediate notification.

## Academic Honesty:

The work you do for this course will be your own, unless otherwise specified. You are not to submit other people's or AI generated work and represent it as your own. I consider academic honesty to be at the core of the University's activities in education and research. Academic honesty is always expected in this course.

## Accommodations for Students with Disabilities

Students with disabilities seeking academic accommodations must first register with the Disability Resource Services (DRS) program, located in MSR 210, ph. (209) 667-3159. Students are encouraged to talk with the instructor regarding their accommodation needs after registering with DRS.

## Students Support Services

Services Director can be accessed via the link: [Student Services | California State University Stanislaus \(csustan.edu\)](#)