

CS 351: Algorithms Syllabus

Fall 2025

Lucas P. Cordova, Ph.D.

Table of contents

1	Syllabus Versioning	3
2	Lecture Information	3
3	Instructor Information	3
3.1	Office Hours	3
3.1.1	Office Hours Schedule	3
4	TA Information	4
4.1	TA Office Hours	4
4.1.1	Office Hours Schedule	4
5	Course Information	4
5.1	Catalog Description	4
5.2	Course Description	4
5.3	Course Learning Outcomes	5
5.4	Textbook & Materials	5
6	Assessments	6
6.1	Grade Weighting	6
6.2	Letter Grade Distribution	6
6.3	Assessment Descriptions	6
6.3.1	Attendance, Participation, Quizzes, & Reading Activities	6
6.3.2	Homework Assignments	6
6.3.3	Programming Projects	7
6.3.4	Student Contributed Lecture	7
6.3.5	Midterm Exam	7
6.3.6	Final Exam	7
7	Course Schedule	7
7.1	Important Dates	7
7.2	Weekly Schedule	7
8	Course Policies	9
8.1	Attendance Policy	9
8.2	Late Work Policy	9



8.2.1	Homework Assignments	9
8.2.2	Programming Projects	9
8.2.3	Participation Activities	10
8.3	Academic Honesty Policy	10
8.3.1	Allowed Collaboration:	10
8.3.2	Prohibited Actions:	10
8.3.3	Programming Projects:	10
9	Willamette Policies	10
9.1	Inclusive Classroom Space	10
9.2	Accessibility and Accommodations	10
9.3	Time Commitment Expectations	11
9.4	Academic Support Resources	11
9.5	Commitment to Positive Sexual Ethics	11
9.6	Land Acknowledgement	11



1 Syllabus Versioning

The syllabus is a living document that may be updated throughout the semester to reflect changes in course policies, schedules, or other important information. The most current version is **1.0 dated 8/25/2025**.

Version	Date	Rationale
1.0	8/25/2025	Initial version created to outline course policies and expectations.

2 Lecture Information

Class Time: MWF 10:20 - 11:20 AM

Location: Smullin B17

3 Instructor Information



Professor: Lucas P. Cordova, Ph.D.

Email: LPCordova@willamette.edu

Office: Ford 210

3.1 Office Hours

Office hours are available by 15-minute [appointments](#) or on a drop-in basis if I am not currently with another student. If the scheduled times do not align with your availability, please don't hesitate to contact me. The scheduling page offers different meeting modalities (in-person, phone, Zoom).

3.1.1 Office Hours Schedule

Day	Time
Monday	1:15 PM - 2:15 PM
Tuesday	10:00 AM - 11:00 AM
Wednesday	1:15 PM - 2:15 PM



Day	Time
Thursday	10:00 AM - 11:00 AM

4 TA Information



TA: Sam Holmes

Email: srholmes@willamette.edu

Location: QUAD

4.1 TA Office Hours

TBD

4.1.1 Office Hours Schedule

Day	Time
TBD	TBD

5 Course Information

5.1 Catalog Description

Advanced study of algorithms and their complexity analysis. Topics include algorithm design paradigms (divide-and-conquer, greedy, dynamic programming), graph algorithms, network flows, and computational complexity theory including NP-completeness. Emphasis on both theoretical understanding and practical implementation.

5.2 Course Description

This course provides a comprehensive study of fundamental algorithms and algorithmic techniques essential for advanced computer science study and professional practice. Students will explore major algorithmic paradigms including divide-and-conquer, greedy algorithms, and dynamic programming, while developing skills in analyzing algorithm efficiency using asymptotic notation.



The course covers essential graph algorithms including shortest paths, minimum spanning trees, and network flows, providing students with tools to solve complex computational problems. Students will also study computational complexity theory, learning to classify problems by their computational difficulty and understanding the significance of P vs NP.

Throughout the course, students will implement algorithms in code, analyze their performance, and apply them to solve real-world problems. Additionally, students will critically analyze and evaluate generative AI solutions for algorithmic problems, developing skills to assess AI-generated code for correctness, efficiency, and appropriateness. By completion, students will be equipped to design efficient algorithms, prove their correctness, analyze their complexity, select appropriate algorithmic approaches for diverse computational challenges, and evaluate AI-assisted solutions in algorithmic contexts.

5.3 Course Learning Outcomes

Upon successful completion of this course, students will be able to:

1. **Design and Analyze Algorithms:** Create efficient algorithms using major design paradigms (divide-and-conquer, greedy, dynamic programming) and analyze their time and space complexity using asymptotic notation.
2. **Implement Advanced Data Structures:** Implement and utilize sophisticated data structures including heaps, hash tables, and graph representations to support algorithmic solutions.
3. **Solve Graph Problems:** Apply fundamental graph algorithms including shortest path algorithms, minimum spanning tree algorithms, and network flow techniques to solve complex problems.
4. **Analyze Computational Complexity:** Classify computational problems by their complexity, understand the significance of P vs NP, and recognize NP-complete problems.
5. **Prove Algorithm Correctness:** Construct formal proofs of algorithm correctness using techniques such as loop invariants and induction.
6. **Select Appropriate Algorithms:** Evaluate and choose the most suitable algorithmic approach for given problems based on efficiency requirements and problem constraints.
7. **Apply Algorithms to Real Problems:** Design and implement algorithmic solutions for practical applications in areas such as network optimization, scheduling, and resource allocation.
8. **Evaluate AI-Generated Solutions:** Critically analyze and evaluate generative AI solutions for algorithmic problems, assessing their correctness, efficiency, and appropriateness for specific use cases.

5.4 Textbook & Materials

- **Primary Text:** *Algorithms Illuminated (Omnibus Edition)* by Tim Roughgarden
 - Available online with interactive features at algorithmsilluminated.org
 - Supplementary video lectures available through the author's website
- **Additional Resources:**
 - Class materials will be provided via Canvas
 - Supplementary readings and research papers distributed via Canvas



- Online algorithm visualization tools and interactive exercises
- **Software Requirements:**
 - Programming environment capable of Python, Java, or C++ development
 - Access to Canvas LMS for assignments and course materials
- **Hardware:** A laptop capable of running development environments, required for in-class activities and exams. If you do not have access to a suitable laptop, please contact me as soon as possible to discuss accommodations.

6 Assessments

6.1 Grade Weighting

Deliverable	Individual Weight	Total Weight
Attendance, Participation, Quizzes & Reading Activities (~15)	1% each (approx.)	15%
Homework Assignments (~10)	2.5% each (approx.)	25%
Programming Projects (3)	10% each	30%
Student Contributed Lecture (group)	-	10%
Midterm Exam	-	10%
Final Exam	-	10%
Total		100%

6.2 Letter Grade Distribution

Grade	Range	Grade	Range
A	92.00	C	72.00 - 77.99
A-	90.00 - 91.99	C-	70.00 - 71.99
B+	88.00 - 89.99	D+	68.00 - 69.99
B	82.00 - 87.99	D	62.00 - 67.99
B-	80.00 - 81.99	D-	60.00 - 61.99
C+	78.00 - 79.99	F	59.99

6.3 Assessment Descriptions

6.3.1 Attendance, Participation, Quizzes, & Reading Activities

Active participation in class discussions, completion of reading assignments from *Algorithms Illuminated*, and engagement with in-class problem-solving activities. These activities cannot be made up if missed.

6.3.2 Homework Assignments

Regular problem sets covering algorithm analysis, complexity theory, and proof techniques. Assignments will include both theoretical problems and algorithm implementation exercises.



6.3.3 Programming Projects

Three major programming projects that require students to implement and analyze significant algorithms:

1. **Divide-and-Conquer Project:** Implementation and analysis of recursive algorithms including mergesort with performance visualization
2. **Dynamic Programming Project:** Complex DP problem implementation (e.g., edit distance, knapsack variations)
3. **Graph Algorithms Project:** Network flow modeling and implementation for real-world scenarios

6.3.4 Student Contributed Lecture

Working in groups of 3, you will have the opportunity to contribute to the course by preparing and delivering a lecture on a relevant topic of your choice. This collaborative exercise encourages deeper exploration of a specific subject, fosters teamwork and collaborative learning, and enhances your presentation skills. Each group is expected to research the topic thoroughly, create an engaging presentation, and lead a class discussion together.

6.3.5 Midterm Exam

Covers material from weeks 1-8, including asymptotic analysis, divide-and-conquer, sorting, basic graph algorithms, and greedy algorithms.

6.3.6 Final Exam

Comprehensive exam covering all course material with emphasis on dynamic programming, advanced graph algorithms, network flows, and computational complexity.

7 Course Schedule

7.1 Important Dates

- **First Day of Classes:** Monday, August 25, 2025
- **Mid-Semester Break:** Friday, October 10, 2025 (no class)
- **Thanksgiving Break:** Wednesday, November 26 - Friday, November 28, 2025 (no class)
- **Last Day of Classes:** Wednesday, December 3, 2025
- **Final Exam:** Tuesday, December 9, 2025, 8:00 AM - 11:00 AM

7.2 Weekly Schedule

The following schedule is **tentative** and may be adjusted based on class progress.

Week	Dates	Topics	Readings	Deliverables
1	Aug 25, 27, 29	Course intro, algorithm fundamentals, Big-O basics	Ch. 1-2	HW1 assigned



Week	Dates	Topics	Readings	Deliverables
2	Sep 1, 3, 5	No class Sep 1 (Labor Day) Asymptotic analysis, notation practice	Ch. 2	Quiz 1, HW1 due, HW2 assigned
3	Sep 8, 10, 12	Divide-and-conquer, recurrence relations, Master Theorem	Ch. 3-4	HW2 due, HW3 assigned, Project 1 assigned
4	Sep 15, 17, 19	Sorting algorithms: merge sort, quicksort, analysis	Ch. 5	Quiz 2, HW3 due, HW4 assigned
5	Sep 22, 24, 26	Graph fundamentals, representations, BFS, DFS	Ch. 7-8	HW4 due, HW5 assigned
6	Sep 29, Oct 1, 3	Greedy algorithms, MST (Kruskal's, Prim's)	Ch. 9-10, 13	Project 1 due , HW5 due, HW6 assigned
7	Oct 6, 8	Dijkstra's algorithm, shortest paths	Ch. 13	Quiz 3, HW6 due, HW7 assigned
8	Oct 13, 15, 17	Dynamic programming introduction and applications	Ch. 16	HW7 due, HW8 assigned, Project 2 assigned
9	Oct 20, 22, 24	Midterm Exam (Oct 22) DP on graphs, Floyd-Warshall	Ch. 17-18	Midterm , HW8 due
10	Oct 27, 29, 31	Network flows, max flow-min cut, Ford-Fulkerson	Supplementary	Project 2 due , HW9 assigned
11	Nov 3, 5, 7	Advanced flow networks, bipartite matching	Supplementary	Quiz 4, HW9 due, Project 3 assigned



Week	Dates	Topics	Readings	Deliverables
12	Nov 10, 12, 14	Computational complexity, P vs NP introduction	Ch. 19, 23	HW10 assigned
13	Nov 17, 19, 21	NP-completeness (Nov 17) Student-led lectures begin (Nov 19, 21)	Ch. 22	Project 3 due , HW10 due
14	Nov 24	Thanksgiving week - Student-led lectures continue	-	Student-led lectures
15	Dec 1, 3	Student-led lectures conclude, Final exam review	-	Student-led lectures, Final exam review

8 Course Policies

8.1 Attendance Policy

Consistent attendance is essential for success in this advanced course. I expect you to attend all classes and actively participate in discussions and problem-solving activities. Attendance will be tracked through participation activities that cannot be made up if missed. If you must miss class due to illness or emergency, please notify me as soon as possible.

8.2 Late Work Policy

Deadlines help ensure your success and prepare you for professional expectations. The following policies apply:

8.2.1 Homework Assignments

- Must be submitted by the designated due date
- Each student receives **three late submission tokens** for the semester
- Each token allows one homework to be submitted late (regardless of days late)
- To use a token, email me before the assignment deadline

8.2.2 Programming Projects

- Late submissions accepted with **10% penalty per day** (including weekends)
- Projects more than 5 days late will not be accepted
- Extensions may be granted for documented emergencies



8.2.3 Participation Activities

- Cannot be made up if missed
- Plan to attend all class sessions

8.3 Academic Honesty Policy

Academic integrity is fundamental to your development as a computer scientist. The following guidelines apply:

8.3.1 Allowed Collaboration:

- Discussing general problem-solving strategies with classmates
- Sharing conceptual insights and debugging approaches
- Working together on understanding course concepts
- Using online resources for general algorithm concepts (with citation)

8.3.2 Prohibited Actions:

- Copying code or solutions from any source
- Sharing your code with other students
- Looking at another student's code or written work
- Using solutions from previous semesters or online sources
- Submitting AI-generated code without significant modification and disclosure

8.3.3 Programming Projects:

- All code must be your original work
- You may discuss algorithms at a high level but must implement independently
- Cite any external resources used for conceptual understanding
- When in doubt, ask for clarification

Violations will result in penalties ranging from assignment failure to course failure, and will be reported to the Academic Standards Committee.

9 Willamette Policies

9.1 Inclusive Classroom Space

I am committed to creating an inclusive learning environment where all students feel welcome and supported. I will gladly honor your request to address you by your affirmed name and pronouns. Please let me know if your preferred name or pronouns differ from those listed in the course roster, and I will make every effort to use them consistently.

9.2 Accessibility and Accommodations

Willamette University is committed to creating accessible learning environments. If you have a documented disability that may impact your performance in this course, please contact Accessible Education Services in Matthews 103 (503-370-6737 or accessible-info@willamette.edu) to discuss accommodations. I am happy to work with you and AES to ensure you have the support needed to succeed.



9.3 Time Commitment Expectations

Following Willamette's Credit Hour Policy, you should expect to spend 6-9 hours per week outside of class on course-related activities. This includes reading, homework, programming projects, and exam preparation. The advanced nature of this course requires consistent effort throughout the semester.

9.4 Academic Support Resources

- **SOAR Center:** Free access to food, clothing, textbooks, and school materials (3rd floor, Putnam University Center)
- **Tutoring Services:** Available through the Academic Resource Center
- **Technical Support:** IT services for software and hardware issues

9.5 Commitment to Positive Sexual Ethics

As a mandatory reporter, I am required to report any incidents of sexual misconduct disclosed to me to Willamette's Title IX Coordinator. For confidential support, contact: - Confidential advocate: confidential-advocate@willamette.edu - WUTalk crisis line: 503-375-5353 - Campus Safety (emergency): 503-370-6911

9.6 Land Acknowledgement

We respectfully acknowledge that Willamette University is located on the ancestral lands of the Kalapuya people, who today are represented by the Confederated Tribes of the Grand Ronde and the Confederated Tribes of the Siletz Indians. We honor their deep connection to this land and recognize the ongoing contributions of Indigenous peoples to our academic community.
