Algorithms
Computer Science 321

Instructor:	Erik Krohn
E-mail:	krohne@uwosh.edu
Text Message:	608-492-1106
Class Time:	Tuesday & Thursday: 3:00pm - 4:30pm
Classroom:	Halsey 309
Office Location:	Halsey 216
Office Hours:	Monday: 11:20am - 12:20pm Tuesday: 2:00pm - 3:00pm Wednesday: 11:20am - 12:20pm Thursday: 2:00pm - 3:00pm
Prerequisites:	CS271 and CS212 with a grade of C or better
Course Website:	`http://canvas.uwosh.edu`
Recommended Textbooks:	*Algorithm Design* by Kleinburg and Tardos. ISBN 9780321295354 *Introduction to Algorithms* by Cormen, Leiserson, Rivest and Stein. ISBN 9780262033848

Course Information
An introduction to the design and analysis of algorithms. Algorithm analysis topics include asymptotic analysis, recurrence relations, loop invariants and amortized analysis. Algorithm design techniques include divide and conquer, the greedy method, and dynamic programming. Additional topics include graph algorithms and NP-completeness.

Course Website
You should check canvas on a regular basis - it will contain lecture notes, handouts, assignments, announcements, and grades. I'll do my best to let you know when something new and important comes up, but it is your responsibility to check the web site frequently for information that you might not get otherwise.

Mini Assignments
You will have daily mini assignments. Mini assignments are generally short and should take less than an hour to complete. You will be assigned a mini assignment every lecture to ensure you are staying current with the material. Mini assignments must be completed in LaTeX and the resulting pdf uploaded to the dropbox before the due date. I will drop your 2 lowest mini assignments. Not all mini assignments will be graded. **No late mini assignments will be accepted.**

Assignments
All assignments requiring "written" work must be written in LaTeX and the resulting pdf submitted electronically via canvas. It is your responsibility to ensure your submission was submitted correctly.

Exams

Exam material will come from the lecture notes, mini assignments, book and assignments. There will be more information about each exam as it approaches. The *tentative* exam dates are listed below. All exams will be taken during the regular class period. These may change, so as the date approaches make sure you've got the most recent information.

- **Exam One** - Tuesday, October 10^{th}, 2023
- **Exam Two** - Tuesday, November 14^{th}, 2023
- **Exam Three** - Thursday, December 14^{th}, 2023

If you are unable to take a scheduled exam, it may be possible to take a make-up exam provided that you do both of the following, which are then subject to my approval:

1. Make arrangements prior to the scheduled exam. For last minute emergencies, telephone me at 424-2063 or leave a message at the computer science office, 424-2069 or send me a text message. No after-the-fact notifications will be accepted.

2. Have a written medical excuse signed by the attending physician OR have a note of justification from the Dean of Students Office.

If allowed, only one make-up exam will be given. It will be a comprehensive exam given at an arranged time during the last week of the semester.

Grading

Course grades will be based on assignments, mini assignments and exams. Your final grade will be computed with the following percentages:

- 45% - assignments
- 15% - mini assignments
- 40% - exams

If you believe anything was graded incorrectly or unfairly and would like to have it regraded, you must let me know about it within *one week* of having the item graded. I will regrade the entire assignment or exam and you may gain or lose points.
Grading will be on a plus/minus system. Grading may be done on a curve depending on the overall performance of the class. If no curve is used, your grade will be computed based on the following:

Percentage	Grade	Percentage	Grade
≥ 92	A	72 - 78	C
90 - 92	A-	70 - 72	C-
88 - 90	B+	68 - 70	D+
82 - 88	B	62 - 68	D
80 - 82	B-	60 - 62	D-
78 - 80	C+	< 60	F

Topic Coverage

- Analysis of algorithms
  - Efficiency
  - Correctness
- Algorithm design strategies
  - Divide-and-conquer
  - Dynamic programming
  - Greedy algorithms
  - Brute-force
  - Backtracking
  - Randomization
  - Approximation
- Algorithms from specific problem domains including:
  - Graph algorithms
  - Numerical algorithms
  - Number theory
  - Geometry
- Data integrity, authentication, and encryption
- Pattern matching
- Parallel algorithms

Learning Outcomes

- The student will be able to prove the correctness of an algorithm using loop invariants and mathematical induction.
- Given a recursive algorithm, the student will be able to examine its recursive structure, determine and mathematically solve the corresponding recurrence relation, and infer the asymptotic runtime of the algorithm using big-O notation.
- The student will be able to use appropriate asymptotic notations for bounding algorithm running times from above and below.
- Given a data structure with an occasional high-cost operation, the student will be able to select an appropriate potential function and use amortized analysis techniques to determine the amortized cost of the operations on the data structure.
- The student will be able to identify problems amenable to divide-and-conquer solutions, derive the details of a solution to such a problem, and analyze the run-time behavior of the corresponding solution.
- Given a problem amenable to a dynamic programming solution, the student will be able to determine the underlying recursion that solves the problem, determine why this recursion is suited to dynamic programming, implement the algorithm using dynamic programming to cache sub-problem solutions, and determine the efficiency of the resulting algorithm.
- The student will be able to identify problems amenable to greedy solutions, derive the details of a solution to such a problem, and analyze the run-time behavior of the corresponding solution.
- The student will be able to identify problems that can be modeled as a graph, select the appropriate graph algorithm to solve the problem and analyze the efficiency of the resulting algorithm.
- The student will be able to identify, from a list of tractable and NP-complete problems, those that are tractable and those that are NP-complete, provide, for each tractable problem, a polynomial-time algorithm that solves it and provide, for each NP-complete problem, a proof that it is NP-complete.
- The student will be able to analyze the run-time behavior of parallel algorithms (e.g., matrix-vector multiplication, parallel merge sort) using the concepts of critical path, speedup, potential parallelism, and parallel slackness within the work/span model.

Other Information

1. Attendance is not taken in this course. However, you are unlikely to do well if you miss lecture and/or lab.

2. **Academic dishonesty** of any kind will not be tolerated. All assignments, labs, mini assignments and exams are to be completed individually. While discussion of ideas and problems with fellow students is encouraged, all projects and labs must be done individually. In certain circumstances, code fragments from the instructor may be provided to eliminate tedious coding or to provide a common framework for all students. **All other code must be original**. Online resources may be used to help you understand the material, but you may not copy online code nor can you "borrow" code from other students, past or present.

   Any suspected academic dishonesty will be dealt with on a case-by-case basis. Any clarification of what does or does not constitute academic dishonesty must take place **before** you turn in questionable work. For clarification on what constitutes academic dishonesty, contact me or consult the printed policy in the UWO Student Discipline Code, Chapter UWS 14.

3. It is the policy and practice of UW Oshkosh to create an inclusive learning environment. If there are aspects of the instruction or design of this course that result in barriers to your inclusion, please notify me as soon as possible. If you do not have an approved accommodation plan, please complete a registration form through Accommodate: `https://uwosh-accommodate.symplicity.com/public_accommodation/`

4. Students are advised to see the following URL for disclosures about essential consumer protection items required by the Students Right to Know Act of 1990: https://uwosh.edu/financialaid/consumer-information/.

5. Diversity drives innovation, creativity, and progress. At the University of Wisconsin Oshkosh, the culture, identities, life experiences, unique abilities, and talents of every individual contribute to the foundation of our success. Creating and maintaining an inclusive and equitable environment is of paramount importance to us. This pursuit prepares all of us to be global citizens who will contribute to the betterment of the world. We are committed to a university culture that provides everyone with the opportunity to thrive.

6. If any substantive changes are made in the course syllabus, such as changes in schedule or assignments, notification will be provided in a timely manner and a revised syllabus will be made available.