CS 321 - Algorithms
Fall 2018
Credits: 3 hours

Instructor: David Furcy **Office:** Halsey 221
Email: furcyd@uwosh.edu **Phone:** 424-1182
Office Hours: **MWF** 10:10-11:10, **TR** 9:30-11:00, or **by appointment**

Class Meetings: **MWF** 9:10-10:10 in Halsey 202

Prerequisites: A grade of C or better in CS 212 and CS 271

Class Web Page: http://www.uwosh.edu/d2l
You should probably check this D2L site more than once a day

Required Text: *Introduction to Algorithms (Third Edition)*
-- T. Cormen, C. Leiserson, R. Rivest, C. Stein (MIT Press)

Tests: Midterm exam: Week of October 29
Final exam: Week of December 10

If you have special needs, please come and talk to me at the end of the first class so I can accommodate your needs right away.

Catalog Course Description: Algorithm design techniques including brute-force, backtracking, divide-and-conquer, dynamic programming and greedy algorithms. Other topics include big-O and amortized analysis, recurrence relations in the analysis of recursive algorithms, numerical algorithms, pattern matching, data integrity, authentication, and encryption.

The following course description is more up to date and is about to be submitted for inclusion in the course catalog.

Proposed Course Description: 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 Outcomes: At the end of the course:

1. The student will be able to prove the correctness of an algorithm using loop invariants and mathematical induction.
2. 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.
3. The student will be able to use appropriate asymptotic notations for bounding algorithm running times from above and below.
4. 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.

5. 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.
6. 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.
7. 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.
8. 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.
9. 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.

Course Grading Policy: Your final grade for this course will be based on 5 components: daily class preparation and participation (CPP), frequent quizzes, homework assignments, a midterm exam, and a final exam. Your overall numerical grade for the course will be computed as the weighted sum of the component grades using the following weights:

Component	Weight
Class Preparation and Participation (CPP)	15%
Quizzes	15%
Assignments	20%
Midterm exam	20%
Final exam	30%

Your final letter grade for the course will be computed using the following mapping:

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

While this overall grading scheme is fixed, I will be happy to discuss any issue you may have with individual grades. If you notice a mistake or have a question regarding a specific grade, please come and talk to me *as soon as possible*. Do not wait until the end of the semester to bring up grading issues. Also, I will *not* be available to discuss grades after the end of the final week.

Attendance and Participation: You are expected to not only attend **every** class meeting but also to come **prepared** for and participate actively in it. Necessary preparation requires you to have studied and assimilated the material covered in previous sessions, to have met with the instructor outside of class to discuss any questions you may have, to have completed the reading assignments, and to have completed all other assignments on time.

Recall that this component of the course accounts for 15% of your grade (much more, really, since it will more or less directly impact your performance in the course as a whole). It will be assessed based on the quality of your answers to daily questions administered via a web-based clicker-like system. These questions will be based on your remembering and understanding of the material in previous reading materials as well as in-class discussions. **Make sure to complete the reading assignment, and to bring your device, EACH DAY**. There is no way to cover all of the materials related to this course in face-to-face meetings. Therefore, **you are responsible for doing your own reading and research**. **Class time will focus on answering your questions, not on lecturing from scratch.**

It is hard to imagine how a student could do well in this course while missing classes, attending them unprepared, or not participating.

On the positive side, I have high expectations for my students and will always support and encourage you. I **strongly encourage** you to **ask any question** or raise any issue you have with the course either during or at the end of class, or during my office hours. I will also gladly meet with you by appointment. Send me email to make an appointment. While I will meet with you as soon as my schedule permits, do not expect me to be widely available before an assignment is due.

Late Submissions: I will describe the submission procedure for your assignments when the time comes. However, let me point out right away that each assignment will come with a deadline (day and time) after which any submission is considered late, **with no exception**. The late-submission policy works as follows. If your submission is past the deadline but still before midnight on the due date, you will lose **20%** of your score. If your submission reaches me after midnight of the due date, I will not grade it and you will receive a **zero**. Late submissions can easily be avoided by starting to work on the assignment right away and asking questions early if you get stuck.

The penalty for late submissions can be waived in **only one** scenario, namely if you give me a signed note from a doctor or a written justification for the extension from the Dean of Students Office. If you miss an exam, you **may** be able to take a make-up exam provided you give me a valid justification (see above) ahead of time if possible. Only one make-up exam will be given. It will be a comprehensive exam scheduled at the end of the semester. If you miss a quiz, you **may** be able to take a make-up quiz, provided you have a valid justification for your absence.

Collaboration versus Cheating: All submissions, unless otherwise stated, must be the work of only one student, namely the one whose name appears on the submission. While it is acceptable and encouraged to discuss the assignments with others, you must submit your own work (or that of your team only, when applicable) unless you can live with a zero and the other potential academic sanctions of cheating. Check out the UWO Student Academic Disciplinary Procedures

(UWS 14) at https://www.uwosh.edu/stuaff/images/Chapter%20UWS%2014.pdf/view for details.

In conclusion, remember that computer science classes require a lot of work in addition to active participation in class. It takes considerable practice to develop the technical and analytical skills targeted by this course. You will need to spend **at least, and typically much more than, three hours of effort outside of class for each in-class hour**. Having said this, I expect every hardworking student to do well in this course.

Have fun this semester and good luck!