

CS 5413
Data Structures and Algorithm Analysis II
Fall 2019
Course Syllabus

Meeting Time and Place: Tue. 4:30 pm - 7:10 am, MSCS 310

Required Textbook: *Introduction to Algorithms, Third Edition* by T. Cormen et al, published by McGraw-Hill

Instructor: N. Park

E-mail: npark(at)cs.okstate.edu

Course Home Page: canvas

Office: MSCS 211

Office Hours: Tuesday 2-4pm

Instructor:

E-mail: (at)cs.okstate.edu

Office: MSCS

Office Phone: TBA

Office Hours: TBA

Prerequisite: CS 3353 (Data Structures and Algorithm Analysis I).

Tentative Grading:

- Programming/Homework Assignments and Quiz: 50%
- Exams (3): 10% each (Tentative Dates: Sep. 10, Oct. 8, Nov. 5)
- Final Exam (Comprehensive): 20% (Date: TBA)

Tentative Grading Scale: [90-100%] A, [80-89%] B, [70-79%] C, [60-69%] D, [0-59%] F

Course Objective and Description: Storage structures, data and information structures, list processing, trees and tree processing, graphs and graph processing, searching, sorting.

Note: Homeworks and program assignments are due at the time and on the date they are due (unless announced otherwise). Late homeworks will **not** be accepted. Late program penalty is **10%** per calendar day, according to the date and time on the printout. Only when verifiable extenuating circumstances can be demonstrated will make-up exams or extended

assignment due dates be considered. Verifiable extenuating circumstances must be reasons beyond control of the students, such as illness or accidental injury. Poor performance in class is not an extenuating circumstance. Advise your instructor of the verifiable extenuating circumstances in advance or as soon as possible. In such situations, the date and nature of the make-up exams and the extended due dates for the assignments will be decided by the instructor.

Attendance Policy: As this is an online course, online attendance is strongly encouraged, but not required or monitored. Students are responsible for any material covered in online-class. Some of the materials covered in class will not be in the textbook. Announcements about homeworks, projects, programming assignments, etc. will be made primarily online and/or by email as necessary. Students are to check their D2L course webpage and emails in their class accounts regularly.

Collaboration Policy: Discussion of techniques and ideas covered in class is encouraged. However, every line of on all assignments must be **your own**.

A statement required by the university: “Care must be taken that exam answers are not seen by others, that term papers or projects are not plagiarized by others or otherwise misused by others, etc. Even passive cooperation in a dishonest enterprise is unacceptable.”

- In **programming assignments**, discussion of techniques in a natural language (such as English) is allowed, but a discussion in a computer or algorithmic language is not allowed. (Computer language discussions and questions are to be limited to the language and should not concern the assignment.) Stealing, giving or receiving any code, drawings, diagrams, texts or designs is not allowed.
- If a proctored-**examination** is given, no discussion of any kind (except with the instructor) is allowed. No access to any type of written material is allowed.
- Students who do not comply with the above described collaboration policy will receive a grade of F in the course. Furthermore, the case will be reported to the University Officials.
- **50% Rule:** For any student to be eligible to receive a passing grade, the student must have earned at least 50% of the total points in each of two areas: (1) programs and other assignments, and (2) tests. Any student whose grades do not satisfy the 50% Rule will automatically receive a grade of F in the course regardless of the total points earned.

Tentative Course Outline:

1. Introduction: basics of data structures and algorithm design/analysis
2. Sorting algorithms and analysis: Heapsort, Quicksort, etc.

3. Trees and Tables: Binary search trees and Hash tables, etc.
4. Dynamic programming technique
5. Greedy algorithms
6. B-tree
7. Binomial heaps
8. Fibonacci heaps
9. Graph theory: fundamentals, minimum spanning trees, shortest path algorithms
10. NP-Completeness