CS 321 Data Structures (Spring 2022)

Instructor: Dr. Jyh-haw Yeh

Office: CCP 247 Phone: 208-426-3034 email: jhyeh@boisestate.edu

URL: http://cs.boisestate.edu/~jhyeh/cs321/cs321.html

Class Time: TueThur 4:30-5:45 PM

Location: CCP 243

Office Hours: TueThur 3:30-4:30 PM and Tue 5:45-6:30 PM

Teaching Assistants:

TAs and Tutors	Office Hours	Location
(TA) Royal Pathak (TA) Chidi Agbo (Tutor) Jonathan Porter	Check tutoring Center Calendar https://www.boisestate.edu/coen-cs/currentstudents/success-tutoring/	CCP 241

Textbook: Introduction to Algorithms, 4th Edition by T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein/McGraw-Hill

Course Objectives: At the end of course, student will be

- able to apply the most efficient known algorithms to solve searching and sorting problems.
- familiar with a variety of different data structures and their appropriate usage.
- able to choose appropriate data structures to implement algorithms.
- able to apply basic graph search algorithms (such as BFS and DFS) to applications.

Catalog Description: Sorting, searching, and order statistics. Further data structures: trees, priority queues, dictionaries, balanced search trees, B-Trees, heaps, hash tables, and graphs.

Pre-requisites: CS 221 Computer Science II, and MATH 189 Discrete Mathematics

Co-requisites: CS-HU 250 Intro. to Version Control, CS-HU 271 Agile Development, and CS-HU 310 Intro to Database System Usage

Course Outline: The following is an estimated schedule. The teaching and learning pace of every semester may be different.

Weeks	Topics	Chapter Readings	Labs, Homeworks & Exams
Week 1	Lecture 1: Getting Started • Insertion sort • Analyzing Algorithms ○ Analyzing insertion sort best, average and worst cases running time	Chapter 2	
	Lecture 2: Programming Assignment #1 Discussion	Programming assignment #1 handout	Assign Programming assignment #1: Performance simulation of 1- and 2-level caches
Week 2	Lecture 1: Sorting algorithms and analysis • Merge sort • Selection sort	Chapter 2	
	Lecture 2: Growth of Functions ● Asymptotic notations: ○ Upper bound (O) ○ Tight bound (Θ) ○ Lower bound (Ω)	Chapter 3	
Week 3	Lecture 1: Cont'd on Growth of function ● Cont'd on asymptotic notations ● Common functions – constants, logarithms, polynomials, exponentials, super-exponentials	Chapter 3	
	Lecture 2: Heapsort ● Heaps ● Maintaining heap property ● Building a heap	Chapter 6	
Week 4	Lecture 1: Heapsort ● Heapsort algorithm ● Priority queue	Chapter 6	
	Lecture 2: Programming Assignment #2 Discussion	Programming assignment #2 handout	Assign Programming assignment #2: simulation of CPU scheduling using a priority queue.
	Lecture 1: Quicksort ● Description of quicksort	Chapter 7	Assign Homework #1 that covers all topics from week 1 to week 5

	● Performance of quicksort ● Randomized quicksort ● Analysis of quicksort		
Week 5	Lecture 2: Sorting in Linear Time ● Lower bounds for sorting ● Counting sort ● Radix sort ● Summary ○ Comparing all sorting algorithms in terms of time complexity, space complexity and stableness	Chapter 8	
Week 6	Lecture 1: Written Homework #1 Discussion (Preparation of Mid-term Exam #1)	Written homework #1 handout	
	Lecture 2: Mid-term Exam 1		Exam #1 covers all topics from week 1 to week 5
Week 7	Lecture 1: Elementary Data Structures ● Stacks, queues and buffers ● Linked lists ○ 8 different linked lists ○ JAVA ArrayList	Chapter 10	
	Lecture 2: Hash Tables ● Direct address tables ● Hash tables ● Hash functions	Chapter 11	
Week 8	Lecture 1: Hash Tables ● Open addressing – linear probing ● Open addressing – quadratic probing	Chapter 11	
	Lecture 2: Hash Tables ● Open addressing – double hashing ● Hash table performance analysis	Chapter 11	

Week 9	Lecture 1: Programming Assignment #3 Discussion	Programming assignment #3 handout	Assign Programming assignment #3: Performance simulation of linear probing and double hashing tables
	Lecture 2: Trees ● Tree concept and terminology, recursive data structures, expression trees	Classnotes (Page 35-38)	
Week 10	Lecture 1: Binary Search Trees ● What is a binary search tree? ● Querying a binary search tree ● Insertion and deletion	Chapter 12	
	Lecture 2: B-Trees ● Definition of B-trees ● Search on B-trees	Chapter 18	
Week 11	Spring Break		
Week 12	Lecture 1: B-Trees ● Insert operation on B-trees ● Delete operation on B-trees	Chapter 18	Assign Homework #2 that covers all topics from week 7 to week 11
	Lecture 2: Programming Assignment #4 Discussion	Programming assignment #4 handout	Assign Programming assignment #4: Bioinformatics – storing/searching human DNA subsequences in a BTree binary file.
Week 13	Lecture 1: Written Homework #2 Discussion (Preparation of Mid-term Exam #2)	Written homework #2 handout	
	Lecture 2: Mid-term Exam #2		Exam #2 covers all topics from week 7 to week 11
Week 14	Lecture 1: AVL Trees ● Definition ● Four rotations ● AVL insert ● AVL delete	Classnotes (page 48-52)	
	Lecture 2: Huffman Trees ● Definition and its data compression application	Class discussion	

	● Huffman tree construction ● Huffman code encoding ● Huffman code decoding		
Week 15	Lecture 1: Elementary Graph Algorithms ● Representations of graphs ○ Adjacency list (Adj-L) ○ Adjacency matrix (Adj-M) ○ Adj-L vs. Adj-M in both time and space complexities ● Breadth-first search	Chapter 20	
	Lecture 2: Elementary Graph Algorithms ● Depth-first search ● Topological sort	Chapter 20	Assign Homework #3 that covers all topics from week 1 to week 16
Week 16	Lecture 1: Minimum Spanning Trees ● Growing a minimum spanning tree ● Kruskal's and Prim's algorithms	Chapter 21	
	Lecture 2: Written Homework #3 Discussion (preparation of Final Exam)	Written homework #3 handout	
Week 17	Final Exam: 5-7 PM on Tuesday (5/03)		Final exam is a comprehensive exam including all topics for the entire semester

Grades and Grading Policies:

Grading:

- Homeworks/Programs: 50%
- Mid-term Exam 1: 15%
- Mid-term Exam 2: 15%
- Final Exam: 20%

Final Grade: You are guaranteed to receive at least the grade as follows (I reserve the right to lower the cutoffs if I feel it is appropriate).

- 87 <= A- < 90 <= A <97 <= A+

- 77 <= B- < 80 <= B < 87
- 67 <= C- < 70 <= C < 77
- 57 <= D- < 60 <= D < 67
- F < 57

Grading Policy:

- Homeworks will not be accepted late.
- Programming assignments must be submitted electronically to the instructor by 11.00PM of the due date to avoid any penalty. Within one week after the deadline, you can still submit your assignment. However, 20% late submission penalty will be applied. No submission will be accepted after one week past the due date.
- All students should submit correct and complete files to the instructor. Any accidentally wrong or incomplete submission may need to submit again and incur the submission penalty. The points you can get for incorrect programs are as follows.
  - Cannot be compiled or run time error: no points.
  - Wrong answer: Varying from 0% to 80% points depends on the answer.

Academic Honesty:

- Each student must work independently unless specified otherwise.
- Determination of academic dishonesty is at the discretion of the instructor of the course within the policy guidelines of the University.