Name	Title	E-mail	Room	Office Hours
Sean Davis	Lecturer	ssdavis@ucdavis.edu	3052 Kemper	MWF 8-11, and by appointment.
Aman Asrani	Reader	arasrani@ucdavis.edu		None
Hongjing Zhang	TA	hjzzhang@ucdavis.edu		None
Huanle Zhang	TA	dtczhang@ucdavis.edu	53 Kemper 55 Kemper	T 6-9pm W 1-3pm

Web page: http://csiflabs.cs.ucdavis.edu/~ssdavis/60/homepage.html

Newsgroup: https://piazza.com/ucdavis/fall2017/ecs60/home

E-mail to Sean should only be regarding personal matters and not questions about assignments or tests, and must come from an ucdavis.edu e-mail account. All course questions should be posted to the piazza newsgroup.

Required Materials: Weiss, Mark Allen, *Data Structures and Algorithm Analysis in C++*, 4th ed., Berkeley, CA: Addison Wesley, 2014.

Prerequisites: ECS 40 or equivalents with grades of C- or better.

Course objectives:

1. Students will be able to apply algorithm analysis to the operations on data structures and interpret the results.
2. Students will be able to understand and evaluate the operations and possible implementations of the following abstract data types: lists, stacks, queues, binary trees, AVL trees, splay trees, tries, B-trees, hash tables, and heaps.
3. Students will understand the operation and characteristics of the following sorting algorithms: insertion sort, shellsort, heapsort, mergesort, quicksort, bucket sort, radix sort, and external sorting.
4. Students will understand the operation and application of graph algorithms including depth first search, breadth first search, shortest path, minimum spanning tree, network flow, and topological sort.
5. Students will be able choose (and justify) appropriate data structures and algorithms for complex programming tasks.

Approximate Course Grading:

Written Homework	15%
Programs (including timetest write-ups)	25%
Two midterms	30%
Final	30%
Discussion attendance/Piazza answers/OH	5% (extra credit)

Letter grades will be approximately: A = 90+% ; B = 80-89%; C = 70-79%; D = 60-69%; F <60%

Work Input/Output

- Written Homework: All homework must be stapled together. Each student must do his or her own work. Written homework should be submitted in the ECS 60 slot in 2131 Kemper by 4:00 PM on the date due.
- Programs: Students should work together in groups of two people to write the programs. The names of both group members must appear in the authors.csv file. All students may help each other with debugging, but each group must write their own code. Program source code will be submitted using the handin facility of UNIX. Each group will submit all of its programs to the handin directory of exactly one of its members. Programs that do not compile will receive no credit. Certain assignments may be submitted to the MOSS program at Stanford for plagiarism analysis. Students that hand in suspicious programs may be reported to Student Judicial Affairs. Each student (not each group) must write his/her own program write-ups for the timetest programs. Students may edit their programs in Sean's office hours, and then have them re-tested. Program edits may only be done within three lectures of the date that program grades are e-mailed. During busy times, a student will only be allowed five minutes to edit their program.
- All work: Late work will **NOT** be accepted without a doctor's excuse. All work will be returned in lecture. All work not picked up in class will be available in my office during office hours. Regrades must be submitted within three lectures of the day the work was first returned to the whole class.

Discussions: M 4:10-5 in 217 Art, T 3:10-4 in 234 Wellman

Exams: Exams are cumulative, closed book, closed notes. The final will be Wednesday, December 13th, 8-10am in the lecture room.

Tentative Schedule

Dates	Subjects	Reading
9/27	Intro, math review, induction.	Chapter 1
9/29	Complexity, ADTs, lists.	Chapter 2, 3.1, 3.2
10/2	Lists cont'd, cursor lists	Handout
10/4	Skip lists, stacks and queues.	Chapter 3.6, 3.7, 10.4.2
10/6	Trees, child-sibling trees	Chapter 4.1
10/9	B-Trees	Chapter 4.7
10/11	B-Trees cont'd	Chapter 4.7
10/13	Trees cont'd: Binary trees, BST, tree traversals.	Chapter 4.2, 4.3, 4.6
10/16	Trees cont'd: AVL trees	Chapter 4.4
10/18	Trees cont'd: Splay trees, amortized analysis intro.	Chapter 4.5, 11.5
10/20	Amortized analysis	Chapter 11.1, 11.5
10/23	Tries, Huffman encoding	Chapter 10.1.2
10/25	Hashing: idea, hash functions, separate chaining	Chapter 5.1-5.3
10/27	Hashing cont'd: open addressing and rehashing	Chapter 5.4-5.5
10/30	Extendible hash, binary heaps, d-heaps	Chapter 5.9, 6.1 – 6.5
11/1	Build heap, disjoint set intro, disjoint set naive data structure	Chapter 8.1-8.4
11/3	Disjoint Set: smart unions, path compression	Chapter 8.6, 9.1, 9.2
11/6	Graph Algorithms: graph definitions, topological sort	Chapter 9.1, 9.2
11/8	Midterm #1, Chapters 1-5, 10.1.2, 10.4.2, 11.5	None
11/13	Graph Algorithms: Critical path analysis, unweighted shortest path	Chapter 9.3
11/15	Graph Algorithms cont'd: Weighted shortest path algorithms	Chapter 9.3
11/17	Graph Algorithms cont'd: Network flow.	Chapter 9.4
11/20	Graph Algorithms cont'd: Minimum spanning tree	Chapter 9.5
11/22	Graph Algorithms cont'd: Articulation points, and catch-up.	Chapter 9.6
11/27	Sorting: insertion sort, lower bound, shellsort,	Chapter 7.1-7.4
11/29	Sorting: heapsort, mergesort, quicksort,	Chapter 7.5-7.7
12/1	Midterm #2, Chapters 1 - 6.5, 8 - 10, 11.5	None
12/4	Sorting: quicksort cont'd, indirect sorting, lowerbound	Chapter 7.7-7.9
12/6	Sorting: bucket sort, radix sort, external sorting.	Chapter 7.10, 7.11.
12/8	NP hard problems, review	

Written Assignment Tentative Due Dates

#	Points	Topic(s)	Due date
1	48	Series, Proofs, and Complexity	10/4
2	10	Lists, Stacks, Queues, Recursion	10/9
3	20	Trees	10/27
4	15	Hash Tables	11/3
5	18	Priority Queues and Disjoint Sets	11/13
6	24	Graph Algorithms	11/29
7	16	Sorting	12/8

Programming Assignments

#	Program Description	**Due date** (11:59pm)
1	Simple ADT Timing Tests, boarding.cpp	10/11
2	BTree insert	10/25
3	Advanced ADT Timing Tests, plus ?	11/6
4	Challenge(s) 1	11/22
5	Challenge 2	12/7