Name	Title	E-mail	Office Hours
Sean Davis	Lecturer	ssdavis@ucdavis.edu	MWF 9-9:50, MW 1:10-4 in 3052 Kemper, and by appointment.
Michael Papadopoulos	TA	mgpapa@ucdavis.edu	W 7-10pm in 53 Kemper
Suraj Kesavan	TA	spkesavan@ucdavis.edu	T 1-2, 4-5 in 55 Kemper, T 3-4 in 53 Kemper
Deepika Chandrasekaran	TA	dchandrasekaran@ucdavis.edu	T 8-10pm in 53 Kemper

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

Newsgroup: https://piazza.com/ucdavis/winter2017/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 20 and 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 212 Wellman, W 4:10-5 in 158 Olson, and F 4:10-5 in 106 Olson.

Exams: Exams are cumulative, closed book, closed notes. The final will be Tuesday, March 21st, 10:30 – 12:30 in 176 Everson.

Tentative Schedule

Dates	Subjects	Reading
1/9	Intro, math review, induction.	Chapter 1
1/11	Complexity, ADTs, lists.	Chapter 2, 3.1, 3.2
1/13	Lists cont'd, cursor lists	Handout
1/18	Skip lists, Stacks and queues.	Chapter 3.6, 3.7, 10.4.2
1/20	Trees, child-sibling trees	Chapter 4.1
1/23	B-Trees	Chapter 4.7
1/25	B-Trees cont'd	Chapter 4.7
1/27	Trees cont'd: Binary trees, BST, tree traversals.	Chapter 4.2, 4.3, 4.6
1/30	Trees cont'd: AVL trees	Chapter 4.4
2/1	Trees cont'd: Splay trees, amortized analysis intro.	Chapter 4.5, 11.5
2/3	Amortized analysis	Chapter 11.1, 11.5
2/6	Tries, Huffman encoding	Chapter 10.1.2
2/8	Hashing: idea, hash functions, separate chaining	Chapter 5.1-5.3
2/10	Hashing cont'd: open addressing and rehashing	Chapter 5.4-5.5
2/13	Extendible hash, binary heaps, d-heaps	Chapter 5.9, 6.1 – 6.5
2/15	Disjoint Set: intro, naive data structures, smart unions.	Chapter 8.1-8.4
2/17	Midterm #1, Chapters 1-5, 10.1.2, 10.4.2, 11.5, 12.3	None
2/22	Disjoint Set path compression, graph definitions, topological sort	Chapter 8.6, 9.1, 9.2
2/24	Graph Algorithms: Critical path analysis, unweighted shortest path	Chapter 9.3
2/27	Graph Algorithms cont'd: Weighted shortest path algorithms	Chapter 9.3
3/1	Graph Algorithms cont'd: Network flow.	Chapter 9.4
3/3	Graph Algorithms cont'd: Minimum spanning tree	Chapter 9.5
3/6	Graph Algorithms cont'd: Articulation points, and catch-up.	Chapter 9.6
3/8	Sorting: insertion sort, lower bound, shellsort,	Chapter 7.1-7.4
3/10	Midterm #2, Chapters 1 - 6.5, 8 - 10, 11.5, 12.3	None
3/13	Sorting: heapsort, mergesort, quicksort,	Chapter 7.5-7.7
3/15	Sorting: quicksort cont'd, indirect sorting, lowerbound	Chapter 7.7-7.9
3/17	Sorting: bucket sort, radix sort, external sorting.	Chapter 7.10, 7.11.

Written Assignment Tentative Due Dates

#	Points	Topic(s)	Due date
1	48	Series, Proofs, and Complexity	1/18
2	10	Lists, Stacks, Queues, Recursion	1/23
3	20	Trees	2/10
4	15	Hash Tables	2/15
5	18	Priority Queues and Disjoint Sets	2/27
6	24	Graph Algorithms	3/8
7	16	Sorting	3/17

Programming Assignments

#	Program Description	**Due date** (11:59pm)
1	Simple ADT Timing Tests, queens.cpp	1/18 (program), 1/23 (paper at 4pm).
2	BTree ranged find	2/1
3	Advanced ADT Timing Tests, plus ?	2/15
4	Challenge(s) 1	3/1
5	Challenge 2	3/16