CS 312 – Algorithm Analysis
Course Syllabus and Calendar – Winter 1999

Instructor: Don Colton

Brigham Young University—Hawaii Campus

Abstract

- **Course Number:** CS 312
- **Title:** Algorithm Analysis
- **Course Description:** (98/98 catalog) Complexity classes, problem-solving strategies, parallel and distributed algorithms, state spaces, search strategies, iterative approximation methods. (Prerequisites: CS 240, 252.)
  (proposed) Analysis of algorithms including searching, sorting, graphs, and trees. (Prerequisite: CS 240, 252) (This course will satisfy both descriptions.)
- **Textbook:** *Introduction to Algorithms*, by: Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. McGraw Hill, 1990. ISBN: 0-07-013143-0.
- **Class Time:** MWF 2:00–2:50 PM
- **Classroom:** GCB 143
- **Instructor (me):** Don Colton
- **My email:** don@colton.byuh.edu
- **My Office:** GCB 130 B, Phone: 293-3478
- **My Office Hours:** MWF 10–11

1 Why Take This Course?

The study of algorithms is focused primarily on speed. One can always buy more memory or a bigger hard disk. It just costs money. One cannot buy time.

The issues with speed revolve around the question of how best to approach the problem. Binary search is a wonderful example. The brute-force lookup method examines each item in a set and stops when the desired item is found. Binary search divides the set into two halves and decides in which half the target would be. Then it repeats this procedure until the set has just one item left. At one second per comparison, and with a set of one million items, the brute-force method would take 11.5 days to find that the target is not in the set. The binary search method would take only about 20 seconds. It is clear to see that even the fastest computer using the brute-force method cannot win against an ordinary computer using the binary search method.

The study of algorithms examines the running time of various programs and looks at some important algorithmic discoveries, such as the divide-and-conquer method used by the binary search. Students will gain skills in both algorithm analysis and algorithm design, and probably gain a few surprising insights along the way.

2 Programming

At this point, it is my intention that most of the programming be done in pseudo-code rather than any particular programming language. We will generally follow the style in the textbook. I may assign some actual programming if I come to believe it would help learning occur better. The best teaching method depends on the actual students taking the class.

3 Course Content

We will cover the first five chapters of the book as background material for what follows. Some of this material is a repeat of what was already seen in CS 235/236 and CS 252, but we will cover it more deeply and rigorously, including attention to proving things. We will spend some time on sorting partly because it is one of the fundamental tools in data processing and doing it well saves lots of time, but mainly to illustrate the analysis of algorithms in a concrete setting. We will cover hashing, with its $O(1)$ set insertion, lookup, and deletion (but rotten successor and predecessor functions). We will cover binary search trees, and in particular red-black trees with their $O(\lg n)$ insertion, lookup, and deletion, plus $O(1)$ successor and predecessor functions. We will look at hybrid data structures that sometimes allow the combining of the best features of different algorithms. We also look at dynamic programming (where the word "programming" is **not** used in the normal computer science sense), greedy algorithms, and spend quite a bit of time on graph algorithms. We will also attend to some special topics: parallel computing, string matching, NP-completeness, and approximation algorithms.

4 Attendance

Due to immigration and veterans requirements the Vice President for Student Life has requested to be notified whenever a student misses four consecutive periods. I comply with this request.

5 Lecture Policies

I typically use a flexible and open lecture style, rather than a regimented sequence of slides. I try to focus on interesting aspects of the subject matter, instead of simply repeating what you have read in the textbook. My goal is that you develop intuition about the subject matter, and get unstuck if you have become stuck. Accordingly, I devote as much time as necessary to answering your questions, especially when those questions seem to be of general interest to the others in the class. (Questions of narrow interest may be deferred to my office.) Otherwise, the time is generally spent in discussion of some topic or other that is closely related to the material in the textbook. I may pose a problem to the class and moderate as we work through it together. This can serve as a jump-start for understanding and mastering new material. My method of teaching is based on the view that learning is a shared activity between the teacher and the student, and that learning proceeds most quickly when interaction occurs.

6 Work Load

In the United States, the expectation for accredited university-level course work is that there be an average of three hours of work per week for every hour of credit awarded. In a lecture class this means one hour in class and two hours outside of class. Some of you are accustomed to working more and others less. It is my goal to keep the work load for an average member of the class at these levels.

You are required to read regularly in the textbook. It is excellent. The authors are highly respected, have a nice presentation style, and they know their material. It is a good book. You will benefit from reading it.

A substantial amount of homework will be assigned. Generally reading and homework together will not exceed two hours per hour in class. If you find the workload to be heavier than that, please let me know.

7 Grading

Grades will be computed on the basis of points earned on reading, homework, quizzes, and tests (two midterms and a final exam).

1/3	reading
1/3	homework and quizzes
1/6	two midterms
1/6	final exam

Reading: Reading is about 462 pages. At 10 pages per hour, it should average a little over an hour per class period. You are permitted to read ahead. Reading is assigned to specific dates by which you must complete it to receive credit. If you read behind the schedule, you will still be able to achieve the learning, but not the points. I assume most of you were going to do the reading on time anyway, so this is a convenient way to give you a small reward for your efforts.

The reading grade is self-reported each day that reading is due. To compute your reading grade, divide the number of pages you read by the pages required, and multiply by the points available. Round to a whole number. Thus if you read about half the pages on a 10-point day, you would report five points earned. (Mostly the number of points is equal to the number of pages.)

Homework: Note: Answer the question that was asked. Occasionally a student will blitz through a question without carefully understanding what is wanted. Generally this results in them answering a much easier question than the one they should. This does not get many points of partial credit.

The primary purpose of homework is to encourage students to master the course material in a lower-stress setting where resources such as the textbook can be consulted in a leisurely way. It is my policy that homework in this course can be done with the aid of other students, and that answers can be compared. Obviously, it is not in anyone's best interest if answers are simply copied from person to person without at least some attempt at understanding. Generally homework means answering questions from the end of each section in the book.

Late Homework: Homework assignments are due at the start of class, and should be turned in to me (at the front of the classroom) when you arrive. Typically I like to discuss a homework assignment on the day that it is turned in, or on the day that I return the graded assignment to the students of the class. This often involves disclosing the answers and discussing how the answers were derived. No late work is accepted **after the homework is discussed in class,** except when I approve it in unusual circumstances.

Quizzes: The primary purpose of quizzes is to measure student learning on a topic-by-topic basis. It shows me how the students are doing, and it shows you where you might need more attention before the exam. (I don't expect to do a lot of quizzes. Each problem takes too long.)

Quiz Makeup: Quizzes cannot be made up except when I approve it in unusual circumstances.

Exams: The primary purpose of examinations (tests) is to gauge student learning by measuring performance in a (possibly timed) supervised situation. It is understood that such a situation creates additional stress for some students. For this reason testing is not used exclusively in the grading process. Each exam will receive a scaled (normalized) score and a letter grade indicating the final course grade that would be earned by consistent performance at the level reflected on that exam. The final counts twice as much as a midterm. **Exam Makeup:** Exams cannot be made up except when I approve it in

unusual circumstances. **Labs (if any):** Each lab counts less than a midterm and involves writing code that implements an algorithm. Bug-free and well-documented code is expected.

Grading Scale: I use the following grading scale, both for individual assignments and for the course as a whole.

93%+	A	90–92.9%	A-	87–89.9%	B+
83–86.9%	B	80–82.8%	B-	77–79.9%	C+
73–76.9%	C	70–72.9%	C-	67–69.9%	D+
63–66.9%	D	60–62.9%	D-	0–59.9%	F

Final Exam: Even though the final exam may count for just a small percentage of your overall grade, you must pass the final exam (60.0% or better) in order to pass the class.

Other notes: I reserve the right to up-scale the scores on any assignment, exam, or whatever, if I feel the absolute numeric grading is too harsh. The transformation may even be non-linear, but in any case, absolute rank order will be preserved and no score will go down.

Whenever you think grading may be unfair or incorrect, I encourage you to discuss specific instances of grading with the me, and to argue for a different grade than I initially assigned. Some very good learning occurs in these settings (for you **and** for me). The best time to do this is during my office hours, or immediately before or after class if the issue is brief. As an alternative, you can submit your argument in writing, together with the original graded work.

Final grades are generally issued by email soon after the final exam, or in-person if I feel that some discussion might be beneficial. Students are invited to visit my office to claim any exams or homework that I am still holding, and to discuss their academic progress. Interim progress reports are issued to the students several times during the course, generally after the midterm exams and before the final exam.

8 Office Hours

Office hours are posted outside my office door. Currently they are Daily 1–2. Office hours are subject to change, as I might discover the need to attend some meeting somewhere, or visit the men's room, or talk to someone in the computer lab.

Students for whom the posted hours are not convenient, or who just want a guaranteed appointment, can come by whenever my door is open (which is most of the time) or contact me by email to make an appointment.

My "open-door policy" is posted on my office door as follows: "If my door is open (even just a bit) feel free to knock and come in. – Bro. Colton"

9 Communication by Email

When I want to say something, or when you want to say something, if we are not in the same room, my first choice is to do it by email. I far prefer it to telephone calls, for instance. When there is an announcement, I will generally tell you in class or send it to you by email. Such announcements might include clarifications on the homework assignments. You are requested to maintain an email account and to provide me with a valid email address.

10 Computer Accounts

As a member of this class, or as a CS major, you are entitled to a computer account in the CS lab. This account gives you access to UNIX systems, software (including compilers and assemblers), email, web surfing, some storage (currently 10 megabytes to start with), and some paper printing (currently 100 pages per CS class). There are also a few modems for dial-in access. If you had a CS account recently, it is probably still active. If not, see me or a lab person (GCB 101) to get set up.

11 Subject to Change

I like to avoid mutual unhappiness, so I avoid changes as much as I can. The course number, title, and description will not change, but I do reserve the right to change anything else in this syllabus at any time for any reason. This includes the grading policies and the course calendar. If you think my changes are unfair, you have the right to complain. As I said, I like to avoid mutual unhappiness, so I avoid changes as much as I can. Any important change will be communicated in class and by email to those affected.

CS 312 Tentative Course Calendar — Winter 1999

mtg	day	date	time	read	Topic	due (pts)
1	Wed	Jan 6	2pm		introduction, syllabus, questionnaire, pretest	
2	Fri	Jan 8	2pm	1	Algorithm, Analysis, Design	read 1-19 (19)
3	Mon	Jan 11	2pm	2	Asymptotic Notation	read 21-37 (17)
4	Wed	Jan 13	2pm	3	Summations	read 42-51 (10)
5	Fri	Jan 15	2pm	4	Substitution, Iteration	read 53-64 (10)
	Mon	Jan 18			**Holiday: Human Rights Day**	
6	Wed	Jan 20	2pm		Master Method	
7	Fri	Jan 22	2pm	5	Sets, Relations, Functions, Graphs, Trees	read 77-96 (20)
8	Mon	Jan 25	2pm	6	Counting and Probability	read 99-111 (13)
9	Wed	Jan 27	2pm	7	Heapsort	read 137-151 (15)
10	Fri	Jan 29	2pm	8	Quicksort	read 153-167 (15)
11	Mon	Feb 1	2pm		Analysis of Quicksort	
12	Wed	Feb 3	2pm	10	Medians and Order Statistics	read 185-191 (7)
13	Fri	Feb 5	2pm	12	Hash Tables	read 219-239 (21)
14	Mon	Feb 8	2pm		Open Addressing	
15	Wed	Feb 10	2pm		**Review for Midterm 1**	Exam (200)
16	Fri	Feb 12	2pm	14	Red-Black Trees	read 263-277 (15)
	Mon	Feb 15			**Holiday: Presidents Day**	
17	Wed	Feb 17	2pm		Red-Black Deletion	
18	Fri	Feb 19	2pm	16	Dynamic Programming	read 299-319 (21)
19	Mon	Feb 22	2pm		LCS	
20	Wed	Feb 24	2pm	17	Greedy Algorithms, Huffman codes	read 329-343 (15)
21	Fri	Feb 26	2pm	18	Amortized Analysis	read 356-374 (19)
22	Mon	Mar 1	2pm	19	B-Trees	read 379-397 (19)
23	Wed	Mar 3	2pm	22	Disjoint Sets	read 440-449 (10)
24	Fri	Mar 5	2pm	23	Breadth First, Depth First	read 463-483 (21)
25	Mon	Mar 8	2pm		Topological Sort	read 485-493 (9)
26	Wed	Mar 10	2pm	24	Minimum Spanning Trees	read 498-509 (12)
27	Fri	Mar 12	2pm	25	Shortest Paths, Dijkstra	read 514-531 (18)
28	Mon	Mar 15	2pm		Bellman-Ford, DAGs, Linear Programming	read 532-543 (12)
29	Wed	Mar 17	2pm		**Review for Midterm 2**	Exam (200)
30	Fri	Mar 19	2pm	27	Maximum Flow	read 579-587 (9)
31	Mon	Mar 22	2pm		Ford-Fulkerson	read 588-604 (17)
32	Wed	Mar 24	2pm	30	Parallel Computers: Pointer Jumping	read 688-708 (21)
	Fri	Mar 26			**Holiday: Kuhio Day**	
33	Mon	Mar 29	2pm		Parallel Computers: CRCW vs EREW	
34	Wed	Mar 31	2pm	33	RSA Public-key cryptosystem	read 831-836 (6)
35	Fri	Apr 2	2pm	34	String Matching: Rabin-Karp	read 853-868 (16)
36	Mon	Apr 5	2pm		String Matching: Knuth-Morris-Pratt, Boyer-Moore	read 869-883 (15)
37	Wed	Apr 7	2pm	36	NP-Completeness: Polynomial Time	read 916-928 (13)
38	Fri	Apr 9	2pm		NP-Completeness: Reducibility	read 929-945 (17)
39	Mon	Apr 12	2pm		NP-Completeness: Reducibility	read 946-960 (15)
40	Wed	Apr 14	2pm	37	Approximation Algorithms: VC, TSP, Set Cover	read 964-978 (15)
41	Fri	Apr 16	2pm		Review for Final Exam	
	Fri	Apr 23	3-6pm		**Final Exam, 3 hours, in class**	Final (400)
	Tue	Apr 27	noon		Final Grades by email (probably)	