CS 301 – Algorithms and Complexity
Course Syllabus and Calendar – Winter 2003

Professor Don Colton

Brigham Young University Hawaii

1 Course Overview

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 each 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.

Prerequisites: Computer Programming III (CS 202) is a prerequisite. For this class we expect you to have programming maturity based on programming experience. Programming III reflects a desired level of maturity.

Discrete Math II (Math 202/L) is a prerequisite. In the discrete math classes you will have learned about trees, graphs, and other data structures and algorithms that are common in Computer Science. When we refer to these same concepts in CS 301, we will expect you to understand them already, or to (re)learn them rapidly.

1.1 The Course

- **Course Number:** CS 301
- **Title:** Algorithms and Complexity
- **Course Description:** Algorithmic analysis, strategies, fundamental algorithms, distributed algorithms, basic computability. (Prerequisites: CS 202, Math 202/L.)
- **Textbook:** *Introduction to Algorithms, 2/e*, by: Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. McGraw Hill, 2001. ISBN: 0-07-013151-1.
- **Class Time:** MWF 2:00–2:50 PM
- **Final Exam:** Fri 18 Apr, 3:00–6:00 PM
- **Classroom:** GCB 153

1.2 The Instructor

- **Instructor (me):** Don Colton
- **My email:** don@colton.byuh.edu
- **My Office:** GCB 130 B
- **Office Hours:** MWF 9 AM to 11 AM

1.3 Office Hours

My office hours are shown above. You can contact me by email to make an appointment at another time. I also have an **open-door policy,** 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"

1.4 Subject to Change

It is possible that I will revise some aspects of the course as we go along. Any changes I make are likely to be to your advantage. If any of my changes seems unfair to you, let me know. I will try to correct it.

2 Course Calendar

Generally the lectures and discussion in class will follow the due dates for the various assignments (shown below).

```
Jan 11: sSeq        sequential search
Jan 14: sBinary     binary search
Jan 16: sSelect     selection sort
Jan 18: sInsert     insertion sort
Jan 23: sBubble     bubble sort
Jan 28: sMerge      merge sort
Feb 01: sHeap       heap sort
Feb 06: sQuick      quick sort
Feb 11: sReport     sort report (term paper)
Feb 13: hash1       hash (single)
Feb 15: hash2       hash (double)
Feb 22: lcs         longest common subseq
Feb 27: bst         binary search tree
Mar 04: dfs         depth first search
Mar 08: bfs         breadth first search
Mar 13: idfs        iterative depth first search
Mar 18: mst         minimal spanning tree
Mar 22: huffman     huffman coding
Mar 29: gColor      graph coloring
Mar 29: msl         maximal sublists
Apr 15: gradplan    graduation plan
Apr 18: final       final exam, in class
        tba-hwq     misc homework & quizzes
        tba-read    misc reading
```

3 Grading

Grades will be computed on the basis of points earned as follows. This may change if, for example, I fail to give enough homework or reading assignments.

150	attendance
150	homework and quizzes
500	programming
100	reading
100	final exam
1000	total

Grading Scale: I use the following grading scale for this class.

930+	A	900–929	A-	870–899	B+
830–869	B	800–829	B-	770–799	C+
730–769	C	700–729	C-	670–699	D+
630–669	D	600–629	D-	0–599	F

Final Exam Score: You must achieve a sufficient score on the final exam, as shown in this table. Your final grade will be the **lower** of your total-points grade (above) and the grade in this table based on your final-exam percentage.

83+	A	80–82	A-	77–79	B+
73–76	B	70–72	B-	67–69	C+
60–66	C	50–59	C-	40–49	D+
30–39	D	20–29	D-	0–19	F

Attendance: I take roll in this class. Attendance counts for 15% of your final grade. Typically attendance is worth 3 or 4 points per day. I take 4-point roll at the start of class. I take 3-point roll about 10 minutes into class. If you come later than that make sure I notice you in class (maybe right after class). Missing and unnoticed persons get zeros.

Due to INS (immigration) and VA (veterans) requirements the Vice President for Student Life is supposed to be notified whenever a student misses four consecutive class days. I try to do this.

In class I follow a general "got questions?" teaching philosophy. It leaves the responsibility for learning with the people that are supposed to learn: the students. (I cannot learn for you.) Canned lectures can be fun and exciting, but frequently the relevant material is in the reading. Our class time will be focused on things you need to do the nearby assignments, or on explaining things that may not be sufficiently clear from the reading.

Reading: I am using *Introduction to Algorithms, second edition* by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. McGraw Hill, 2001. ISBN: 0-07-013151-1.

I will occasionally assign chapters to be read. The chapters will also be discussed in class.

Labs/GradeBot: Some of your time will be spent writing programs. They will be graded by my robotic grader, GradeBot.

GradeBot is my robotic program grader. It is generally available 24 hours a day, seven days a week, to grade and return your lab assignments. This is currently done via web, "turnin," or email.

For grading, GradeBot is correct and authoritative. It is your boss. It is your client. It is your Drill Sergeant. There is always a particular correct behavior that it demands. You must make your program behave in exactly the way that GradeBot is requiring (including spelling errors, if any). Be sure to look at a sample "conversation" with GradeBot before you start writing your program.

If you discover a case where you believe that GradeBot is wrong, tell me about it. If you found an error in GradeBot, I generally reward you with some extra credit. Otherwise, you must assume GradeBot is right.

4 Course Content

The CS 301 course covers the following CC2001 Knowledge Units. These are defined in Computing Curricula 2001, a joint project of IEEE-CS and ACM. The IEEE Computer Society and the Association for Computing Machinery are the two major professional societies in computer science.

AL1. Basic algorithmic analysis

- Asymptotic analysis of upper and average bounds
- Differences among best, average, and worst case
- Big O, little o, Ω (omega), and Θ (theta) notation
- Standard complexity classes
- Empirical measurements of performance
- Time and space tradeoffs in algorithms
- Using recurrence relations to analyze algorithms

AL2. Algorithmic strategies

- Brute-force algorithms
- Greedy algorithms
- Divide-and-conquer
- Backtracking
- Branch-and-bound
- Heuristics
- Pattern matching and string/text algorithms
- Numerical approximation algorithms

AL3. Fundamental computing algorithms

- Simple numerical algorithms
- Sequential and binary search algorithms
- Quadratic sorting algorithms (selection, insertion)
- $O(n \lg n)$ sorting algorithms (quick-, heap-, merge-)
- Hash tables, including collision-avoidance strategies
- Binary search trees
- Representations of graphs (adjacency list, matrix)
- Depth- and breadth-first traversals
- Shortest-path algorithms (Dijkstra and Floyd)
- Transitive closure (Floyd's algorithm)
- Minimum spanning tree (Prim and Kruskal)
- Topological sort

AL4. Distributed algorithms

- Consensus and election
- Termination detection
- Fault tolerance
- Stabilization

AL5. Basic computability

- Tractable and intractable problems
- The halting problem

5 Special Needs

Brigham Young University Hawaii is committed to providing a working and learning atmosphere, which reasonably accommodates qualified persons with disabilities. If you have any disability that may impair your ability to complete this course successfully, please contact the students with Special Need Coordinator, Leilani A'una at 293-3518. Reasonable academic accommodations are reviewed for all students who have qualified documented disabilities. If you need assistance or if you feel you have been unlawfully discriminated against on the basis of disability, you may seek resolution through established grievance policy and procedures. You should contact the Human Resource Services at 780-8875.

6 Preventing Sexual Harassment

Title IX of the education amendments of 1972 prohibits sex discrimination against any participant in an educational program or activity that receives federal funds, including Federal loans and grants. Title IX also covers student-to-student sexual harassment. If you encounter unlawful sexual harassment or gender-based discrimination, please contact the Human Resource Services at 780-8875 (24 hours).