



CS 312, Section 001, Spring 2007: Algorithm Analysis

Instructor: Brian Price

Web: <http://rivit.cs.byu.edu/~bprice/>

Email: bprice AT rivet DOT cs DOT byu DOT edu

Phone: 801-422-2347

Office: 3308 TMCB

Office Hours: to be announced

Lecture: MWF 12:00-1:50pm, 368 MARB

TA: Nathan Purser

Email:

Office Hours:

Location: CS 312 TA Desk in TMCB Basement

TA: Luther Tychonievich

Email:

Office Hours:

Location: CS 312 TA Desk in TMCB Basement

Web-site: <http://rivit.cs.byu.edu/~bprice/CS312/>

Announcements: See the BYU BlackBoard page for this course. Please check for announcements daily.

Last updated: *2007 April 27*

Description and Objectives

Welcome to CS 312! This course will provide an introduction to the analysis of algorithms from both a theoretical and a practical perspective. Algorithms solve problems, and we will explore a wide variety of problems, some abstract and some down-to-earth. Application domains include image processing, text sorting, gene sequence alignment, 3-D graphics, and logistical planning. As such, CS 312 provides an excellent mathematical foundation and will serve as a gateway to problem solving in other domains.

We will explore several families of algorithms, with the ability to solve problems of increasing complexity:

- Divide and Conquer
- Graph Algorithms
- Greedy Algorithms
- Dynamic Programming
- Real-valued Linear Programming
- Branch and Bound (including A*)
- Probabilistic

The theoretical analysis of algorithms can answer questions that are difficult or impossible to answer by other means. However, one's trust in theoretical results should be tempered by a firm understanding of their limits. The most important strength and limitation of theoretical analysis is the generality one can achieve. This generality is not always useful when attempting to solve a small set of problems with a well-defined performance goal.

After completing CS 312, the student will have learned to successfully apply the following theoretical analysis techniques:

- Asymptotic analysis
- Probability theory in service of average case analysis
- Recurrence relations for the analysis of recursive algorithms

The practice and theory of algorithms covered in this class will serve you well as you move on to more ambitious subjects in Computer Science, and it will serve you well in your careers. This course is also required by many of the department's 400-level courses.

As a practical objective of the course, CS 312 students will use Visual Studio, the C# programming language, and the .NET framework. This gives CS 312 students solid exposure to a widely used development environment. We find that many CS graduates working in industry use Microsoft tools in their professions, and their experience in CS 312 will serve them well.

Web site

Class web site: <http://rivit.cs.byu.edu/~bprice/CS312/>

Class forum: Blackboard

Online grade book: Blackboard via RouteY

The class web site contains all of the pointers you might need, including a link to the detailed, day-to-day schedule and a link to the forum on Blackboard. The forum allows me to post announcements and pointers as they come up. The forum also allows you to comment on those announcements and pointers. To post a comment you will need to authenticate with Route Y. I'll use the Blackboard forum to post useful tips for using Visual Studio and to alert you to updates to the schedule and homework assignments as the semester progresses.

Please actively read the class website for announcements and pointers.

Textbook

The textbook for the course is Algorithms by Dasgupta, Papadimitriou, and Vazirani. We also use supplemental handouts. We hope you'll find these resources useful. If not, we hope you'll tell us why not.

Development Tools

Visual Studio is installed on the Windows machines in department labs, and you can get a free copy if you are a BYU CS student through the MSDN Academic Alliance here:

BYU: <http://csalliance.cs.byu.edu/>

Microsoft: <http://msdn.microsoft.com/academic/>

You must use Visual Studio to develop in C# on the Windows platform to complete the projects.

To help you get started, here are some useful articles at MSDN:

- "Getting Started with Visual C#": <http://msdn2.microsoft.com/en-us/library/a72418yk.aspx>
- "C# for C++ Developers": <http://msdn2.microsoft.com/en-us/library/yyaad03b.aspx>
- "Information for Java Programmers": <http://msdn2.microsoft.com/en-us/library/ms228394.aspx>

Out-of-class Learning Experiences

Interviews with students suggest that how and what students do outside of class is the strongest predictor for engagement and learning in a given class.¹ The same study suggests that working in groups was also well-correlated with successful engagement and learning in class. So you are welcome to work in groups outside of class. However, you should each write and turn in your own work.

Projects: There will be six projects throughout the semester: five full projects and one lighter project.

Each of the full projects is split into two parts. In the first part of the full projects and for the light projects, you implement an algorithm in such a way as to meet a conservative performance requirement. A reasonable implementation will sail through the performance requirement. You will prepare a report describing the empirical analysis of your algorithm, and you will answer questions posed in the project guidelines.

In the second part of the full projects, you go a little further by thinking of some way to improve your algorithm. In CS, “improve” means to make the algorithm perform better according to some metric, including running time, memory consumption, or applicability to more problems. In 312, we’ll also include “has a nice visualization tool for observing the behavior of the algorithm as it runs” to our list of improvements. We leave the definition of an improvement open. If you want to do something not listed among the suggestions that will accompany each project guideline document then tell us what your plan is, and go for it (highly encouraged!). You will satisfy the second part by writing a 1-2 page report that describes your improvement, your objective for the improvement, and an empirical analysis from which you conclude whether or not you accomplished your objective.

We expect that you’ll get an idea for your improvement as you work on getting your program to go fast enough to pass off. Remember, the performance requirement to pass off your program is pretty conservative. Just get it running correctly and fast enough, turn it in and start working on your improvements. Getting it to run fast enough and writing your report will be about 75% of the total work for a project, and doing the improvement and writing the improvement report will constitute about 25% of the total work.

The light projects will entail a lighter workload and include no improvement component. The light projects come at the end of the schedule and should allow you time to prepare well for the final exam.

Homework: Homework is due on paper at the beginning of class on the due date specified by the schedule. Homework will be announced in class and posted linked to the schedule. We will have homework nearly every day. Think of the homework as a way to solidify concepts learned during the lecture and

¹ Light, Richard J, “Making the Most of College: Students Speak their Minds” Harvard University Press, 2001.

from the required reading. Often the homework will help you prepare to succeed on your projects as well. Late homework is not accepted. A reasonable attempt at a homework assignment will earn at least ½ credit.

As is the case with the projects, you are encouraged to work together on the homework, but you should each write up and turn in your own homework and indicate with whom you worked.

Exams: There will be one mid-term exam and a final exam. The mid-term exam will take place in the testing center with a three hour time limit. You may bring one page of hand-written (by you) notes into each exam. The exams will be available from Thursday through Saturday to allow for schedule flexibility. You are encouraged to study together in preparation for the exams.

The final exam will be held in class according to the University's calendar (see the course schedule). You may also bring one page of hand-written (by you) notes into this exam. As per University policy, no exceptions will be granted to final exam time or location. Plan your travel accordingly.

Course Policies

Grade policy: Your overall grade will consist of the following components:

7%	Project #1
3%	Improvement #1
7%	Project #2
3%	Improvement #2
7%	Project #3
3%	Improvement #3
7%	Project #4
3%	Improvement #4
3%	(Light) Project #5
10%	Project #6
10%	Homework
17%	Mid-Term Exam
20%	Final Exam

Grade scale: The following table shows the guaranteed lowest final grade for any given percentage (i.e., if you earn 93.0% of the possible points above, you will get a final grade no lower than an A). I reserve the right to adjust final grades upward in your favor at the end of the semester in order to account for shortcomings in my effort to relay concepts to you.

Grade	Minimum Percent
A	93.0
A-	90.0
B+	87.0
B	83.0
B-	80.0
C+	77.0
C	73.0
C-	70.0
D+	67.0
D	63.0
D-	60.0
E	0.0

The University policies on I (Incomplete) grades will be strictly followed.

Late policy: Homework assignments are not accepted late, as they are meant to help you prepare for the lectures. You either did it or you didn't.

The projects are due at 5pm on the specified due date. For these project assignments, each student has a budget of 5 free late days (weekdays) which can be spent at any time, except that no more than 3 may be used on any single project. This should allow for some flexibility in the syllabus, given a student's other class constraints and unexpected life events. Once the free late day budget for an assignment is spent, each additional late day results in a penalty of 10% per day, until that assignment is handed back by the instructor or TA (at which point no credit is possible). Each late day ends at 5pm.

Weekends and University holidays will not be counted as late days.

All work must be submitted by the last day of instruction for the Fall semester.

In this class, our early and late policies should allow you enough flexibility for most special circumstances during the semester. Additionally, ***special circumstances can be accommodated by prior arrangement***. If you anticipate you need some kind of accommodation to handle a special circumstance for which our early and late policies will not suffice (e.g., for homework), talk to your instructor as soon as possible. In all but the most dire circumstances, further special circumstances will not be accommodated after the fact. For example, if you are planning to get married during the semester, arrange with your instructor to make up missed work, extend deadlines etc. If you got married the third week of classes and missed a project, but did not make prior arrangements, then no accommodations will be made. If you were in a car accident and at the Emergency Room and could not make prior arrangements, then of course we can make accommodations.

Cooperation policy: You are encouraged to work together on your homework, project reports, and improvements. If you work with someone else, simply indicate who in your final report. While you each need to write your own code

and report, you can both work on the same improvement and you can share results. For example, you might think up a clever but difficult improvement. You might each only be able to code up half of it. You might combine your implementations into one and write a report about that. Note that each of you wrote about half the code and each of you wrote your own report but you only ended up with one program in the end.

Students are welcome (and encouraged!) to discuss the general topics of the class, including details of specific algorithms or methods appearing in the lectures or readings. Use of the Blackboard forum for this purpose is strongly encouraged.

Please consult the sections on Projects and Homework for additional policy on cooperation.

Honor: The Honor Code includes a statement of standards regarding academic honesty. Academic honesty includes writing your own programs, properly citing sources in reports and doing your own work on tests. Examples of academic dishonesty include sharing code for projects with other students, turning in someone else's writing as your own report and cheating on an exam. The first violation of academic honesty standards will result in your course grade being lowered 1 grade level and you will be required to either redo the work or receive a 0 on the assignment. The second violation will result in failing the class. All violations of academic honesty are documented and reported to the Honor Code office.

Harassment: Harassment of any kind is inappropriate at BYU. Specifically, BYU's policy against sexual harassment extends not only to employees of the university but to students as well. If you encounter sexual harassment, gender-based discrimination, or other inappropriate behavior, please talk to your professor, contact the Equal Employment Office at 422-5895 or 367-5689, or contact the Honor Code Office at 422-2847.

Disabilities: BYU is committed to providing reasonable accommodation to qualified persons with disabilities. If you have any disability that may adversely affect your success in this course, please contact the University Accessibility Center at 422-2767. Services deemed appropriate will be coordinated with the student and instructor by that office.