COMP 157 – Design & Analysis of Algorithms  
(3 credits)  
Fall 2009

This syllabus describes the way this course will be conducted. Please contact the instructor if there are statements that are unclear, or that you disagree with. Questions and feedbacks are ALWAYS welcomed. Please feel free to talk to me if you have any concerns or needs. Specifics are subject to change with appropriate notice.

Instructor: Dr. Jinzhu Gao

- **Office:** CSB 110   **Phone:** 946-3037   **FAX:** 946-7326   **Email:** jgao@pacific.edu
- **Homepage:** http://www1.pacific.edu/~jgao
- **Lecture:** MWF 2:00 pm – 2:50 pm at WPC 131
- **Office hours:**
  - MWF: 3:00pm – 4:30pm
  - Or by appointment

Textbooks:


Sakai:

Sakai, [https://pacific.rsmart.com](https://pacific.rsmart.com), will be used for distributing course notes, assignments and announcements. Students are responsible for checking the site regularly.

Prerequisite:

COMP53 and COMP47. Students entering COMP157 should be competent programmers with a working understanding of:

- object-oriented design and programming
- discrete mathematics
- common data structures including: arrays, stacks, queues, lists, maps, hash tables, dictionaries, trees and graphs
- techniques for analyzing and comparing the efficiency of algorithms
- iteration and recursion as algorithm development techniques
- common algorithms such as: updating structures, search and selection, sorting
**Course Description:**

This course is an advanced study in the design, analysis and application of algorithms. The emphasis will be on understanding how to select, evaluate and use the common algorithms which have been developed by previous generations of computer scientists. We will organize our study by design techniques, meaning that we will look at groups of algorithms that utilize similar strategies for addressing common problems.

While the primary emphasis of this course is on developing an abstract theoretical understanding of algorithms, practical usage is also important. For this reason, some of the homework assignments may require you to design, implement and debug programs.

**Course Objectives:**

By the end of the course, you will become skilled at:

- Using mathematics to determine the efficiency of algorithms.
- Understanding the common strategies for algorithm development.
- Selecting and utilizing a wide variety of common algorithms for designing problem solutions.
- Understanding why some algorithms are very hard and defy analysis.
- Transforming generic algorithm descriptions into working programs to solve specific problems

**Course Topics:**

- **Efficiency analysis techniques:** Methods for determining the efficiency of an algorithm and for determining the best possible efficiency for particular problems.
- **Brute force algorithms:** These algorithms are based on a "try everything and see what works" approach.
- **Divide and conquer algorithms:** These algorithms first solve smaller instances of a problem, then combine the solutions to solve the complete problem.
- **Decrease and conquer algorithms:** These algorithms exploit the relationship between a problem and a smaller instance of the same problem.
- **Transform and conquer algorithms:** These algorithms work by transforming a problem into a problem that is more readily solvable.
- **Space and time trade-offs:** These algorithms use more memory to yield more time-efficient solutions.
- **Dynamic Programming:** These algorithms are used to solve problems that have overlapping subproblems by solving each subproblem once and recording the solutions in a manner that can be used to construct a solution to the complete problem.
- **Greedy algorithms:** These algorithms make decisions based on what appears to be the best next step at each stage in the solution process.
- **Iterative improvement algorithms:** These algorithms start with a feasible solution and then improve it in small steps until reaching a solution that cannot be improved.
- **Very hard problems:** Some problems cannot be solved efficiently by any algorithm. These are the well-known P, NP and NP-complete classes of problems. However, there are
strategies for dealing with this problems, such as looking for approximate solutions, or designing algorithms that will generally find a solution in a reasonable time.

**Grading:**

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework</td>
<td>50%</td>
</tr>
<tr>
<td>Attendance and Participation</td>
<td>10%</td>
</tr>
<tr>
<td>Midterm Exams</td>
<td>20%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>20%</td>
</tr>
</tbody>
</table>

**Homework:** There will be regular homework assignments, some of which will include substantial programming exercises.

When paper submission is requested, everything you submit must be typeset and printed. No handwritten submissions will be accepted, except for rare cases where an explicit exception is made in the assignment specification.

When electronic submission is requested, follow submission guidelines carefully. All files submitted (including all program code) should have your name and the assignment number clearly indicated near the top of the file.

All programs submitted must *compile and execute*. Zero credit will be given for programs that do not meet this minimum standard.

You are free to use any programming language you like for programming exercises.

**Exams:** There will be two midterm exams, and a comprehensive final exam. The midterm exams will be held during class.

**Attendance and Participation:** Class attendance and participation is necessary and expected. Participation requires that you are properly prepared for classroom discussions and activities, and have completed all reading assignments before the relevant class.

Students missing a class are responsible for making up the material discussed in that class on their own. Students are responsible for being aware of any announcements made during their absence.

**Late Assignments and Make-up Work:** The only acceptable excuses for missing an assignment due date are serious illness, death in the immediate family or important professional activities. Illness or death in the family may require documentation. Excuses for professional activities must be approved by the instructor in advance.

**Individual Work and Collaboration:** Computer professionals usually work in a cooperative environment, yet proper assessment requires that work be done by individuals. To alleviate confusion, the following policy will be followed:

Collaborative work is encouraged. This includes students working together on problem sets, planning solution strategies and helping each other to debug programs. Collaboration must stop short of the writing of program code or English that represents your work. You may not directly copy the work of another student. It is your responsibility to ensure that the work you submit is an honest representation of your own understanding of the material.

Marginal cases will be resolved by oral examination of the students involved. If they understand the material in the assignment, it will be considered honest collaboration. If they do not, then it will be considered academic dishonesty.

**Honor Code and Academic Dishonesty:** The University Honor Code is an essential element of academic integrity. It is a violation of the Honor Code to give or receive information from another student
during an examination or to submit all or part of someone else's work as one's own. If a student violates the Honor Code, the faculty member may refer the matter to the Office of Student Life. If found guilty, the student may be penalized with failure of the assignment or the course. The student may also be reprimanded or suspended from the University.

Cases of academic dishonesty will not be tolerated. On a first offense, the student will be given a written warning and a grade of zero for the work in question. On a second offense, notice will be sent to the student's academic advisor, the Dean of the School of Engineering and Computer Science and the Dean of the student's own school, and the matter will be turned over to the Office of Student Life for resolution.