# Computer Science / Mathematics 340
## Analysis of Algorithms

**Instructor**: *Steven Lindell* Office: Link 308
(610) 896-1203 slindell@haverford.edu

**Course Description**: Qualitative and quantitative analysis of algorithms and their corresponding data structures from a precise mathematical point of view. Performance bounds, asymptotic and probabilistic analysis, worst case and average case behavior. Correctness and complexity. Particular classes of algorithms such as sorting searching will be studied in detail.

**Schedule & location**: MWF 2:15-3:45 Stokes 14

**Consultation hours**: MW 1:00-2:00 and also by appointment.

**Text**: *Introduction to the Design and Analysis of Algorithms* by Anany V. Levitin (Villanova University), 3rd edition ©2012. ISBN: 0-13-231681-1

**Supplementary reading**: *Algorithm Design* Jon Kleinberg and Eva Tardos
*Introduction to Algorithms* Corman, Leiserson, Rivest

**Prerequisites**: One year of introductory CS (or comparable experience programming), and discrete mathematics 231 (or comparable experience with proofs).

**Grading**:

| | | |
|---|---|---|
| *Homework* | 20% | (about ten assignments) |
| *Midterm* | 30% | (one exam) |
| *Final* | 40% | (cumulative) |
| *Participation* | 10% | (during lectures and discussions) |

**Homework**: Weekly exercises. Your work should be neatly and clearly presented, as these are *very* important in the grading (and learning) process. The solutions you hand in should be understandable to someone who knows the material, but not necessarily your approach. Not doing (or at least trying) all the homework is a good recipe for a poor (or failing) grade. No unexcused late homework will be accepted, but your lowest assignment will be dropped.

**Problem section**: Attendance is required at the once weekly Friday discussion section where we will go over solutions and provide approaches for the homework problems.

**Rules and regulations**: Everything turned in for a grade must be your own work, although collaboration on problems is strongly encouraged (especially working in groups). Spoken and scribbled *ideas* on how to solve homework problems may be exchanged, but *not* detailed written solutions. You must write up your own solutions and should acknowledge your collaborators. You may use the Internet *only if the problem's hint suggests that you do so*, but you still must cite the source and put it into your own words. There is no assistance or help allowed on the examinations. Digital devices are not allowed in the classroom, unless you are taking notes with a stylus, or have permission because of an accommodation (see below).

**Special accommodations**: Students who think they may need accommodations in this course because of the impact of a disability are encouraged to meet with me privately early in the semester, and should also contact the Office of Disabilities Services to verify their eligibility for reasonable accommodations as soon as possible.

# Revised Syllabus

| Week | Date | Topic | Assignment |
|---|---|---|---|
| **1** | **9/4** | **LABOR DAY** | |
| | 9/6 | Introduction to complexity and correctness | Section 1.1 |
| | *9/8* | *The Euclidean algorithm* | Section 1.2 |
| 2 | 9/11 | Review of fundamental algorithms: sorting and searching | Section 1.3 |
| | 9/13 | Review of fundamental data structures: lists, graphs, trees | Section 1.4 |
| | *9/15* | *In-class exercises* | *Homework #0* |
| 3 | 9/18 | Orders of growth | Sections 2.1 - 2 |
| | 9/20 | Analysis of algorithms; recurrences | Sections 2.3 - 4 |
| | *9/22* | *Solutions #0* | *Homework #1* |
| 4 | 9/25 | selection sort; string matching; closest pair; convex hull; | Sections 3.1 - 3 |
| | 9/27 | assignment problem; breadth-first and depth-first search | Sections 3.4 - 5 |
| | *9/29* | *Solutions #1* | *Homework #2* |
| 5 | 10/2 | Topological sorting | Section 4.2 |
| | 10/4 | Selection and median finding; binary search trees | Sections 4.4 - 5 |
| | *10/6* | *Solutions #2* | *Homework #3* |
| 6 | 10/9 | Master method; mergesort and quicksort | Sections 5.1 - 2 |
| | 10/11 | Binary tree traversals; closest pair and convex hull | Sections 5.3, 5.5 |
| | *10/13* | *Solutions #3* | *Homework #4* |
| | **10/16-20** | **FALL BREAK** | |
| 7 | ~~10/23~~ | ~~Gaussian elimination~~ | ~~Section 6.2~~ |
| | 10/25 | Height-balanced search trees | Section 6.3 |
| | *10/27* | *Solutions #4; Review for midterm* | *Homework #5* |
| 8 | 10/30 | Heap sort; Horner's rule | Sections 6.4 - 5 |
| | 11/1 | Binary exponentiation; counting paths in graphs | Sections 6.5 - 6 |
| | *11/3* | *Solutions #5* | *Midterm* |
| 9 | 11/6 | Counting and Radix sorts | Section 7.1 |
| | 11/8 | Open and closed Hashing | Section 7.3 |
| | *11/10* | *Discuss midterm solutions* | *Homework #6* |
| 10 | 11/13 | Memory functions; coin changing | Section 8.2 |
| | 11/15 | Warshall's and Floyd's algorithms for transitive closure | Section 8.4 |
| | *11/17* | *Solutions #6* | *Homework #7* |
| 11 | 11/20 | Longest Common Subsequence, and Higman's Lemma | |
| | 11/22 | No class | |
| | ***11/24*** | ***THANKGIVING*** | |
| 12 | 11/27 | Minimum spanning tree (Prim's and Kruskal's algorithms) | Sections 9.1 - 2 |
| | 11/29 | Activity selection; shortest paths (Dijkstra's Algorithm) | Sections 9.2 - 3 |
| | *12/1* | *Solutions #7* | *Homework #8* |
| 13 | 12/4 | Maximum matching in bipartite graphs | Section 10.3 |
| | 12/6 | Maximum network flow | Section 10.2 |
| | *12/8* | *Solutions #8* | *Homework #9* |
| 14 | 12/11 | Lower bounds for sorting and searching | Sections 11.1 - 2 |
| | 12/13 | Tractability: P, NP, co-NP, completeness | Section 11.3 |
| | *12/15* | *Solutions #9; Solutions #10; Review for final* | *Homework #10* |
| | **12/22** | **End of exam period** | **Final Exam** |