EECE 537: Foundations of Computing
http://www.eece.unm.edu/~dbader/eece537/

Instructor: Dr. David A. Bader, EECE 230B, 277-6724, dbader@eece.unm.edu
Office Hours: Monday/Wednesday 8:30-9:30AM, and by appointment
Textbook: T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein,

Course Description: Computational aspects of engineering problems. Topics include machine models and computability, classification, and performance analysis of algorithms, advanced data structures, approximation algorithms, introduction to complexity theory and complexity classes.
Prerequisite: EECE 331.

Grading:

(20 %) Exam I
(20 %) Exam II
(20 %) Exam III
(10 %) Homework
(25 %) Project
( 5 %) Class participation

Topics:

- Proof Techniques
- Introduction to Models of Computation
- Solution of Recurrence Relations
- Algorithmic Analysis
- Graph Algorithms
- Complexity Theory
- Approximation Algorithms
CLASS POLICIES

1. **Homework.** There will be approximately eight (8) homework assignments.

2. **Exams.** All exams will be take-home exams, but **NO COLLABORATION IS ALLOWED.** Specifically, you may not communicate with any person about any aspect of an exam until after the hand-in deadline, even if you have already handed in your exam. You may use your notes from the course and any textbooks. Please reference all sources that you use. You will be given several days to work each exam. The exam dates for this semester are as follows.

   - **Exam I:** Sept. 24-26
   - **Exam II:** Oct. 22-24
   - **Exam III** Dec. 5-7

3. Please let me know as soon as possible if you will need to re-schedule an exam, or have any special needs during the semester.

4. **Independent Research Project.** For this project, you will choose a problem, describe and analyze its known sequential algorithms, provide a detailed literature survey, and if appropriate, implement the algorithm. A list of possible topics is included in this syllabus, or you may research your own topic with my approval. By **September 24, 2001,** you must email to me your problem topic and several bibliographic references on the problem. The project is due **December 7, 2001.** You must submit a hardcopy to me, and email the \( \text{T\LaTeX} \) source files and any code you have written.

   You are required to use \( \text{T\LaTeX} \) to prepare your project report (see the “Information for Students” web page.) Please choose a style (or class) file from the “Information for Students” web page, for example, a UNM Report, IEEE, or ACM.

   Your project will be graded on several criteria including the difficulty of your chosen topic, improvements you may discover to the algorithms, your understanding of the material, the quality and completeness of your presentation including bibliography, the accuracy of your theoretic analyses, and the significance of your empirical study. Plagiarism of any kind will not be tolerated.

5. **Class e-Mailing List.** You are required to subscribe yourself to the class mailing list via the web page [http://www.eece.unm.edu/mailman/listinfo/eece537/](http://www.eece.unm.edu/mailman/listinfo/eece537/) This list will be used for important course information.
Example Problems for Independent Project

The following is a list of possible topics for your independent project:

- Exact algorithms for Traveling Salesperson
- Approximation algorithms for Traveling Salesperson
- Shotgun sequencing of DNA
- Ear decomposition of a graph
- Inversion distance between two signed permutations
- Sorting an unsigned permutation by reversals
- Exemplar distance between two signed permutations
- Transposition distance between two signed permutations
- Perfect matching in a graph
- Range search data structures
- Nearest neighbor search (Kd-trees)
- Point location
- Polygon partitioning
- Delaunay triangularization of planar point sets
- 3-d Convex hulls
- Non-uniform Mesh partitioning
- Maximum flow
- Graph colorability
- Suffix trees
- Approximation algorithms for bin packing
- Facilities location problem
- Approximation algorithms for knapsack
- Approximation algorithms for vertex cover
- Comparing heaps (e.g. binomial, Fibonacci, skew)
- Euler path
- Bipartite graph matching
- Empirical tests of splay trees versus binary search trees
- DNA sequence alignment
- Randomized algorithms
- Union-find data structures and analysis
- Strong encryption algorithms
- Computational geometry
- Computational biology
- Vertex updating problem of a minimum spanning tree
- Hierarchical clustering
- String matching
- Stable marriage problem
- Random sampling techniques
- Byzantine agreement
- Maximal independent sets
- Randomized sorting
- Steiner tree problems
- Longest common subsequence/substring
- Chinese postman problem
- Voronoi diagrams of point sets
- Multiple string comparison
- Maximum likelihood phylogenetic techniques
- Maximum parsimony phylogenetic techniques
- Unrooted tree enumeration
- Neighbor-joining tree reconstructions