

Spring 2005
ECE / CS 509
Tuesday/Thursday 9:30 - 10:45 AM
Classroom: ECE Room 310

ECE/CS 509: Parallel Algorithms

<http://www.ece.unm.edu/~dbader/ece509/>

Instructor: Dr. David A. Bader, ECE 230B, 277-6724, dbader@ece.unm.edu

Office Hours: Tuesday/Thursday 8:00-9:00 AM, or by appointment

Textbook: JáJá, *Introduction to Parallel Algorithms*

Course Description: Design and analysis of parallel algorithms using the PRAM model, with emphasis on graph algorithms, searching and sorting, and linear algebra applications. Embedding into hypercubic and related networks. Introduction to parallel complexity theory.

Grading:

- (30 %) Midterm
- (30 %) Final
- (25 %) Independent Project
- (10 %) Homework
- (5 %) Class participation

CLASS POLICIES

1. This course covers theoretical foundations of parallel algorithms and not parallel programming¹. The emphasis of this course is on the design and analysis of parallel algorithms.
2. Class announcements will be sent to the **CSE509** mailing list. To subscribe, visit:
<http://www.ece.unm.edu/mailman/listinfo/cse509/>.
3. **Homework.** There will be approximately eight (8) homework assignments. You may discuss homework problems in depth with your colleagues, but must turn in your own work.
4. **Exams.** The midterm exam is scheduled for 10 March 2005. The final exam is scheduled for 5 May 2005.
5. **Independent Project.** For this project, you will choose a problem, describe and analyze known sequential and parallel algorithms, and provide a detailed literature survey.
6. 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.

¹ECE 432 / CS 442 covers practical aspects of parallel computing: architectures, languages, and programming.

INDEPENDENT PROJECT DETAILS

Please choose an advanced topic in parallel algorithms for your project, for instance, but not limited to, sections 4.3.2, 5.3, and sections from chapters 6, 7, 8, 9, and 10, from JáJá's text, or topics in other texts listed under **Parallel Algorithms** on the "Information for Students" web page <http://www.ece.unm.edu/~dbader/links.html>

If you wish to implement your algorithms using shared-memory or message-passing programming, **you are responsible for all aspects of using the parallel machine.** If you need access to a parallel computer, please see the above [web page](#) for accessing parallel computers at UNM's Center for High Performance Computing, which also contains [documentation](#) for parallel programming. If you do not already have an account on the parallel machine, get one as soon as possible.

The use of \LaTeX to prepare your project report (see the "Information for Students" web page.) is required. Please choose a style (or class) file from the "[Information for Students](#)" web page, for example, a UNM Report, ACM, IEEE, or AHPCC. The teTeX distribution of \LaTeX is available for all UNIX (Solaris, Linux, FreeBSD) platforms; check with your system administrator. The MiKTeX distribution is available for Microsoft platforms. If you would like a CD-R containing MiKTeX , emacs, ghostscript and gv, and ispell, please give me a blank CD-R with your name on it.

As a suggestion, you may want to follow this outline for your project report.

1. **Introduction** to your algorithm, including a detailed literature survey;
2. **Sequential algorithms;**
3. **Parallel algorithms;** Describe and cite any previous literature on this parallel algorithm. Describe complexity, speedup, and efficiency, for each algorithm, and comparisons between the algorithms. Use figures as necessary.
4. **Experimental Methodology;** Describe the parallel machine and parallel programming environment. Describe the experiments you will use to benchmark your algorithm. You will need to run a variety of problem sizes on a collection of machine configurations.
5. **Experimental Results;** Report (graphically or in tabular format, as appropriate) your experimental results from running the experiments. Do the empirical execution times match your parallel algorithmic complexity? Is the algorithm obtaining the expected parallel speedups and efficiencies? Why or why not?
6. **Conclusions;** Is there a gap between the theoretical parallel algorithms and the parallel implementation?
7. **References**

(Due 10 February 2005) Proposal. This draft details the problem you wish to use for the project and its current best sequential algorithm. Include any necessary literature survey and a self-contained description of the problem. Please identify the parallel platform and programming environment you plan to use.

(Due 5 May 2005) Final Report. Submit a hardcopy of your final report and email me the follow items (as email attachments):

1. electronic copy of the source files (\LaTeX , images, bib, etc.) which can regenerate your project report,
2. the sequential and parallel codes, and
3. input files necessary to re-test your algorithm.

ECE/CS 509 COURSE OUTLINE

I. Introduction (Ch.1)

- A. Parallel Processing
- B. Background
- C. Parallel Models
- D. Performance of Parallel Algorithms
- E. The Work-Time Presentation Framework
- F. The Optimality Notion

II. Basic Techniques (Ch.2)

- A. Balanced Trees
- B. Pointer Jumping
- C. Divide and Conquer
- D. Partitioning
- E. Pipelining
- F. Accelerated Cascading
- G. Symmetry Breaking

III. Lists and Trees (Ch.3)

- A. List Ranking
- B. Euler-Tour Techniques
- C. Tree Contraction
- D. Lowest Common Ancestors

IV. Searching, Merging, and Sorting (Ch.4)

- A. Searching
- B. Merging
- C. Sorting
- D. Sorting Networks
- E. Selection

V. Graphs (Ch.5)

- A. Connected Components
- B. Minimum Spanning Trees
- C. Biconnected Components
- D. Ear Decomposition
- E. Directed Graphs

VI. Realistic Models of Parallel Computation

- A. Bulk Synchronous Parallel (BSP)
- B. LogP
- C. Shared-Memory (SMP)

VII. Clusters of SMPs

- A. Communication Primitives
- B. Sorting
- C. 2D FFT