

CS483 Design and Analysis of Algorithms

Lecture 1 Introduction and Prologue

Instructor: Fei Li

lifei@cs.gmu.edu with subject: CS483

Office hours:

Room 5326, Engineering Building, Monday 2:00pm - 4:00pm or by
appointments

Course web-site:

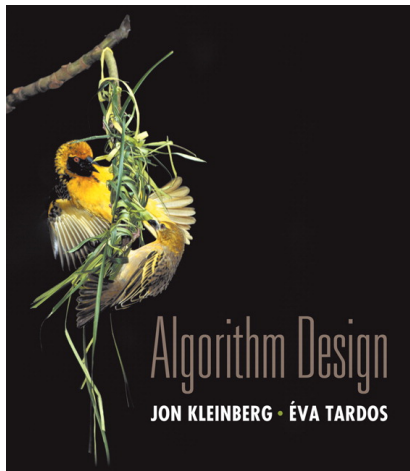
http://www.cs.gmu.edu/~lifei/teaching/cs483_spring12

About this Course

- ▶ About this Course
(From 2007-2008 University Catalog) Analyze computational resources for **important problem types** by alternative algorithms and their associated data structures, using **mathematically rigorous techniques**. **Specific algorithms analyzed** and improved
- ▶ Prerequisites
CS310 (Data Structures) and CS330 (Formal Methods and Models) and MATH125 (Discrete Mathematics I)
- ▶ **Weekly Schedule**
 - ▶ When: **Monday 4:30 pm – 7:10 pm**
 - ▶ Where: **Robinson Hall B203**
- ▶ Attendance is expected.

Required Textbooks

1. **Algorithm Design** by Jon Kleinberg and Eva Tardos



How to Reach Me and the TA

1. Instructor: **Fei Li**
2. Email: **lifei@cs.gmu.edu**
3. Office: Room 5326,
Engineering Building
4. Office hours: Monday
2:00pm - 4:00pm or by
appointments

1. Teaching Assistant:
Katherine Russell
2. Email: **krusselc@gmu.edu**
3. Office:
4. Office hours: Thursday
3:00pm - 5:00pm

Making the Grades

1. Your grade will be determined 45% by the **take-home assignments**, 20% by a **midterm exam**, and 35% by a **final exam**
2. Tentatively, there will be 9 assignments; each assignment deserves 5 points
3. Hand in hard copies of assignments in class. **No grace days for late assignment**. All course work is to be done independently. Plagiarizing the homework will be penalized by maximum negative credit and cheating on the exam will earn you an F in the course
4. Class participation is useful for borderline cases.
5. Tentative grading system:
A (≥ 85), B ($\in [70, 85)$), C ($\in [60, 70)$), D ($\in [50, 60)$),
and F (< 50)

Any Questions?

Course Outcomes

1. An understanding of classical problems in Computer Science
2. An understanding of classical algorithm design and analysis strategies
3. An ability to analyze the computability of a problem
4. Be able to design and analyze new algorithms to solve a computational problem
5. An ability to reason algorithmically