BBM 202 - ALGORITHMS

HACETTEPE UNIVERSITY
DEPT. OF COMPUTER ENGINEERING

INTRODUCTION

Acknowledgement: The course slides are adapted from the slides prepared by R. Sedgewick and K. Wayne of Princeton University.

Instructor and Course Schedule

- Section I - Dr. Aykut ERDEM
  - aykut@cs.hacettepe.edu.tr
  - Office: 111

- Section II - Dr. Erkut ERDEM
  - erkut@cs.hacettepe.edu.tr
  - Office: 112

- Section III - Dr. Adnan Ozsoy
  - adnan.ozsoy@hacettepe.edu.tr
  - Office: 114

- Lectures: Monday, 09:00 - 09:45 @D1-D2-D3
  Tuesday, 09:00 - 10:45 @D1-D2-D3
- Practicum (BBM204): Friday, 14:00-16:50@D3-D4-D8

Instructor and Course Schedule

- Teaching Assistants
  - Selim Yilmaz selimy@cs.hacettepe.edu.tr
  - Levent Karacan karacan@cs.hacettepe.edu.tr
  - Merve Özdes merve@cs.hacettepe.edu.tr

- Practicum (BBM204): Friday, 14:00-16:50@D3-D4-D8
This course concerns programming and problem solving, with applications. The aim is to teach students how to develop algorithms in order to solve the complex problems in the most efficient way. The students are expected to develop a foundational understanding and knowledge of key concepts that underlie important algorithms in use on computers today. The students are also expected to gain hands-on experience via a set of programming assignments supplied in the complementary BBM 204 Software Practicum. Grading for BBM 204 will be based on a set of quizzes (20%), and 4 programming assignments (done individually) (80%).

Why study algorithms?

Their impact is broad and far-reaching.

- **Internet**: Web search, packet routing, distributed file sharing, ...
- **Biology**: Human genome project, protein folding, ...
- **Computers**: Circuit layout, file system, compilers, ...
- **Computer graphics**: Movies, video games, virtual reality, ...
- **Security**: Cell phones, e-commerce, voting machines, ...
- **Multimedia**: MP3, JPG, DivX, HDTV, face recognition, ...
- **Social networks**: Recommendations, news feeds, advertisements, ...
- **Physics**: N-body simulation, particle collision simulation, ...

Why study algorithms?

- **Old roots, new opportunities**.
  - Study of algorithms dates at least to Euclid.
  - Formalized by Church and Turing in 1930s.
  - Some important algorithms were discovered by undergraduates in a course like this!

Why study algorithms?

To solve problems that could not otherwise be addressed.

**Ex.** Network connectivity.
Why study algorithms?

For intellectual stimulation.

“For me, great algorithms are the poetry of computation. Just like verse, they can be terse, allusive, dense, and even mysterious. But once unlocked, they cast a brilliant new light on some aspect of computing.” — Francis Sullivan

“It has often been said that a person does not really understand something until he teaches it to someone else. Actually a person does not really understand something until he can teach it to a computer, i.e. express it as an algorithm. The attempt to formalise things as algorithms lead to a much deeper understanding than if we simply try to comprehend things in the traditional way. Algorithm must be seen to be believed.” — Donald Knuth

Why study algorithms?

They may unlock the secrets of life and of the universe.

Computational models are replacing mathematical models in scientific inquiry.

\[
\begin{align*}
E &= mc^2 \\
F &= ma \\
\left[-\frac{\hbar^2}{2m}\nabla^2 + V(r)\right] \Psi(r) &= E \Psi(r)
\end{align*}
\]

20th century science (formula based)

21st century science (algorithm based)

“Algorithms: a common language for nature, human, and computer.” — Avi Wigderson

Why study algorithms?

To become a proficient programmer.

“I will, in fact, claim that the difference between a bad programmer and a good one is whether he considers his code or his data structures more important. Bad programmers worry about the code. Good programmers worry about data structures and their relationships.” — Linus Torvalds (creator of Linux)

“Algorithms + Data Structures = Programs.” — Niklaus Wirth

Why study algorithms?

For fun and profit.
Why study algorithms?

- Their impact is broad and far-reaching.
- Old roots, new opportunities.
- To solve problems that could not otherwise be addressed.
- For intellectual stimulation.
- To become a proficient programmer.
- They may unlock the secrets of life and of the universe.
- For fun and profit.

Communication

- The course webpage will be updated regularly throughout the semester with lecture notes, programming assignments and important deadlines.

Getting help

- Office Hours
- BBM204 Software Practicum
  - Course related recitations, practice with algorithms, etc.
- Communication
  - Announcements and course related discussions

Coursework and grading

Class participation/Attendance 5%
- Contribute to Piazza discussions.
- Attend and participate in lecture.

Midterm exams 55% (30+25)
- Two closed-book exams
  - in class on March 20 and April 17

Final exam 40%
- Closed-book
- Scheduled by Registrar.
BBM204 Software Practicum

Programming assignments (PAs)
- Four assignments throughout the semester.
- Each assignment has a well-defined goal such as solving a specific problem.
- You must work alone on all assignments stated unless otherwise.

Important Dates
- Programming Assignment 1    27 February
- Programming Assignment 2    20 March
- Programming Assignment 3    17 April
- Programming Assignment 4    1 May

Cheating

What is cheating?
- Sharing code: by copying, retyping, looking at, or supplying a file
- Coaching: helping your friend to write a programming assignment, line by line
- Copying code from previous course or from elsewhere on WWW

What is NOT cheating?
- Explaining how to use systems or tools
- Helping others with high-level design issues

Penalty for cheating:
- A violation of academic integrity, disciplinary action

Detection of cheating:
- We do check
- Our tools for doing this are much better than most cheaters think!

Resources (textbook)


Booksite.
- Brief summary of content.
- Download code from book.

https://algs4.cs.princeton.edu/home/

Course outline

Introduction
Analysis of Algorithms
- Computational Complexity

Sorting
- Elementary Sorting Algorithms,
- Mergesort,
- Quicksort,
- Priority Queues and HeapSort

Searching
- Sequential Search
- Binary Search Trees
- Balanced Trees
- Hashing,
- Search Applications
<table>
<thead>
<tr>
<th>Course outline</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Graphs</strong></td>
</tr>
<tr>
<td>• Undirected Graphs,</td>
</tr>
<tr>
<td>• Directed Graphs,</td>
</tr>
<tr>
<td>• Minimum Spanning Trees,</td>
</tr>
<tr>
<td>• Shortest Path</td>
</tr>
<tr>
<td><strong>Strings</strong></td>
</tr>
<tr>
<td>• String Sorts, Tries,</td>
</tr>
<tr>
<td>• Substring Search,</td>
</tr>
<tr>
<td>• Regular Expressions,</td>
</tr>
<tr>
<td>• Data Compression</td>
</tr>
</tbody>
</table>