Description

The main motivation of this course is to look closely at some popular applications and identify and understand the core algorithmic ingredients. So we consider applications such as Google, Cryptography, Data Compression, Web Crawling, Mapquest, Routing Protocols, Akamai, Airline Scheduling, Pattern Matching and Amazon, discuss simple combinatorial abstractions of the underlying problems in these applications, and the clever algorithmic techniques and data structures that are needed to solve these abstractions - the algorithms we will discuss are not necessarily exactly what these applications use, but will serve to illustrate the general concepts. The applications, problems and algorithms are chosen so that (i) they are popular, and (ii) they can be used to illustrate as many different algorithmic design paradigms as possible, and highlight the utility of algorithms in everyday applications.

The course consists of three modules: Graph Theory, Number Theory and Linear Algebra, and we will discuss algorithmic problems in each of these modules. The end goals of the course are: (i) Understanding of different algorithmic paradigms, (ii) Introduction to formal analysis of algorithms, (iii) Exposure to modeling complex applications in real life combinatorially, and (iv) Development of problem solving skills.

Pre-requisites

- CS 2604, Data Structures and File Management, or CS 2606, Data Structures & Object Oriented Development II
- MATH 3134, Applied Combinatorics and Graph Theory, or MATH 3034, Introduction to Proofs.

Grading Policy

Grading for the course is on a 1000-point scale, with the points distributed as follows:

- Homework assignments: 5 at 120 points each 600
- Midterm exam: March 20, 150
- Final exam: May 9, 1:05-3:05 PM 250

A typical homework assignment would consist of 4-5 problems. Submission in LaTeX is strongly recommended. **No late homework will be accepted.**

Ethics

The Honor Code applies. All work submitted must be the student’s own work. Students may solicit help only from the instructor or the GTA.
Announcement

If any student needs special accommodations because of a disability, please contact the instructor during the first week of classes.

List of Topics

The course would consist of the following topics.

1. **Web Crawling**: Graph search algorithms (BFS, DFS). Sections 3.1, 3.2, 3.3 from [1]. This topic will serve as an introduction to graph algorithms and basic analysis methods.

2. **Mapquest, Google/Routing**: Dijkstra’s algorithm. Section 4.4 of [1]. This topic will illustrate the greedy paradigm, and will recap priority queues.

3. **Airline scheduling**: Introduce network flows and discuss the scaling algorithm. Also discuss additional applications of network flows. Sections 7.1, 7.2, 7.3 and 7.9 of [1].

4. **Cryptography**: Discuss algorithms for bitwise multiplication, modular exponentiation, primality (a restricted version of Rabin-Miller), and the RSA public key system. Based on Sections 1.1, 1.2, 1.3, 1.4 of [2]. This topic would introduce "algorithms on numbers", some basic number theory, and a basic randomized algorithm. We will also discuss the Faster Integer Multiplication algorithm using Divide-and-Conquer, based on Section 2.1 of [2], which will illustrate the Divide-and-Conquer paradigm.

5. **Data Compression**: Huffman’s algorithm, and a brief discussion of Lempel-Ziv and JPEG. Based on Sections section 4.8 of [1] and the notes by Luca Trevisan. This topic would illustrate the greedy paradigm.

6. **Google ranking**: The link analysis algorithm (also Hits), based on [3]. This topic would introduce some basic linear algebra. Some brief discussion on sampling/streaming techniques to deal with large data sets.

7. **Google News**: We will formulate this as a clustering problem and will discuss Kruskal’s algorithm for minimum spanning trees and a hierarchical clustering algorithm using this. Based on Sections 4.5, 4.6, 4.7 of [1]. This topic would also discuss the greedy paradigm.

<table>
<thead>
<tr>
<th>Dates</th>
<th>Topics</th>
<th>Textbook Sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>01/16</td>
<td>Introduction</td>
<td></td>
</tr>
<tr>
<td>01/18</td>
<td>Basics of Graph Theory</td>
<td>Section 3.1 of [1]</td>
</tr>
<tr>
<td>01/23, 01/25</td>
<td>Searching in graphs</td>
<td>Sections 3.2, 3.3 of [1]</td>
</tr>
<tr>
<td>01/30, 02/01</td>
<td>Shortest Paths</td>
<td>Section 4.4 of [1]</td>
</tr>
<tr>
<td>02/06 - 02/20</td>
<td>Cryptography</td>
<td>Chapter 1 of [2]</td>
</tr>
<tr>
<td>02/22 - 03/03</td>
<td>Clustering and spanning trees</td>
<td>Sections 4.5 - 4.7 of [1]</td>
</tr>
<tr>
<td>03/20 - 03/27</td>
<td>Data Compression</td>
<td>Section 4.8 of [1]</td>
</tr>
<tr>
<td>03/29-04/12</td>
<td>Network Flows</td>
<td>Sections 7.1-7.3 and 7.9 of [1]</td>
</tr>
<tr>
<td>04-17 - 05/01</td>
<td>Google Ranking and social networks</td>
<td>[3]</td>
</tr>
</tbody>
</table>

Table 1: Rough schedule of classes. The schedule is likely to change depending on the needs.
References

