20 CS 471: Design & Analysis of Algorithms I

When: TBD
Instructor: Aravind Ranganathan
Office: 805B Rhodes
Email: rangana@mail.uc.edu
Office Hours: TBD, Teaching Assistant: TBD

Web:
All necessary course information, announcements, lecture notes, assignments / solutions, and other relevant materials will be posted online. Students are expected to visit the webpage for any announcements. Note: All posted materials will be in pdf format.

Grading Policy:
There will be 8 weekly quizzes, 4 homework assignments, 1 midterm and 1 final exam. The final grade distribution will be computed as follows:
- Quiz (Best 6 out of 8) : 30%
- Homework Assignments (Best 3 out of 4) : 30%
- Midterm Exam : 20%
- Final Exam : 20%

There will only be letter grades for the course and will be computed as follows: 90-100% = A; 80-89% = B; 70-79% = C; 60-69% = D; 59% and below = F. The average, maximum and minimum grades from the midterm and the final exam will be posted.

There will be a quiz every Friday at the beginning of the class for 10 minutes and will be on the contents covered in that week. There will be absolutely no re-quizzes. All homework assignments are to be done individually without any group discussion; students are to contact the instructor or the TA for doubts. The date for turning in the homework assignments will be announced along with the homework; no homework will be accepted after the deadline.

Note: Any kind of plagiarism in homework or exams will directly result in an immediate "F" grade for the course; the student will also not be allowed to retake the course in future.

Attendance & Classroom Behavior:
Attendance is not mandatory, but students are strongly advised to not miss classes.

Description:
- Review of sequential algorithms.
- Major design strategies: the greedy method, divide-and-conquer, dynamic programming
- Graph and network algorithms.
- Parallel algorithms and architectures.
- Distributed and Internet algorithms.
- Introduction to lower bound theory.
- Introduction to NP-completeness.
- Approximation algorithms.
Textbook:

Prerequisites by topic:
1. Knowledge and experience in a high-level programming language.
2. Familiarity with abstract data types and data structures.
3. Familiarity with mathematical concepts from calculus, linear algebra, discrete mathematics, and elementary probability theory.
4. Course in algorithms including following topics:
   - Measuring the complexity of algorithms: Best-case, Average, and Worst-case complexities, asymptotic notation
   - Design and analysis of classical algorithms such as Binary Search, Insertion Sort, etc.
   - Design and analysis of sequential sorting algorithms, e.g., Mergesort, Quicksort, Heapsort, Treesort, Radix Sort.
   - Mathematical properties of binary trees: lower bounds for depth and leaf-path length.
   - Induction and correctness proofs.
   - Recurrence relations.
   - Implementing graphs and digraphs; DFS and BFS search and traversals & applications to shortest paths in unweighted graphs and topological sort of dags (directed acyclic graphs)

Approximate Schedule of Topics (by week):
1. Fundamentals of Sequential Algorithms (Chapters 1-4).
2. Fundamentals of Sequential Algorithms (continued)
3. The Greedy Method (Chapter 12)
   - Data Compression and Huffman Codes, Knapsack Problem.
4. Divide-&-Conquer (Chapter 13)
   - Polynomial and Large Integer Multiplication, Strassen’s Matrix Multiplication, Fast-Fourier Transform
5. Dynamic programming (Chapter 14)
   - Parenthesization of Matrix Products, Optimal Search Trees.
6. Graph and Network Algorithms (Chapters 8, 12, 14)
7. Graph and Network Algorithms (continued)
8. Internet Algorithms (Notes)
   - Web Searching, Crawling and Ranking
9. Parallel Algorithms and Architectures (Chapters 5 and 6)
   - Intro to Lower Bound Theory and NP-completeness (Chapter 10 & 20)